kilobaud™ MICROCOMPUTING

February 1979
$2.50

✓ *Reader Service—see page 179*

Considering a Microcomputer?

Be Sure to Check Out the Product Offerings of the World's Largest Full Line Microcomputer Company.

All Ohio Scientific machines come with microcomputing's fastest full feature BASIC-in-ROM or on-Disk for instant use.

	Minimum Configuration	Base Price
Challenger I Series		
Economical computer systems that talk in BASIC. Ideal for hobbyists, students, education and the home.		
Superboard II — World's first complete system on a board including keyboard, video display, audio cassette, BASIC-in-ROM and up to 8K RAM	4K RAM	$ 279
Challenger IP — Fully packaged Superboard II with power supply	4K RAM	$ 349
Challenger IP Disk — Complete mini-floppy system expandable to 32K RAM	16K RAM	$1190
Challenger IIP Series		
Ultra high performance BUS oriented microcomputers for personal, educational, research and small business use.		
C2-4P — The professional portable	4K RAM	$ 598
C2-8P — The world's most expandable personal machine for business or research applications	4K RAM	$ 799
C2-4P Disk — The ultimate portable	16K RAM	$1464
C2-8P Single Disk — Ideal for education, advanced personal users, etc.	16K RAM	$1738
C2-8P Dual Disk — Most cost effective small business system	32K RAM	$2597
Challenger II Serial Interface Series		
Same great features as Challenger IIP Series for those who have serial terminals: small business, education, industry.		
C2-0—Great starter for users with a terminal	4K RAM	$ 298
C2-1 — Great timeshare user accessory; cuts costs by running simple BASIC programs locally	4K RAM	$ 498
C2-8S — Highly expandable serial machine, can add disks, etc.	4K RAM	$ 545
Challenger III The Ultimate in Small Computers		
The unique three processor system for demanding business, education, research and industrial development applications.		
C3-S1 — World's most popular 8″ floppy based microcomputer	32K RAM dual floppys	$3590
C3-OEM — Single package high volume user version of C3-S1	32K RAM dual floppys	$3590
C3-A — Rack mounted multi-user business system directly expandabe to C3-B	48K RAM dual floppys	$5090
C3-B — 74 million byte Winchester disk based system. World's most powerful microcomputer	48K RAM dual floppys	$11,090

OHIO SCIENTIFIC also offers you the broadest line of expansion accessories and the largest selection of affordable software!

Compare the closest Ohio Scientific Model to any other unit you are considering. Compare the performance, real expansion ability, software and price, and you will see why we have become the world's largest full line microcomputer company.

I'm interested in OSI Computers. Send me information on:

☐ Personal Computers ☐ Small Business Computers
☐ Educational Systems ☐ Industrial Development Systems
☐ I'm enclosing $1.00 for your 64-page small computer buyer's guide.
Ohio residents add 4% tax.

Name ______________________
Address ______________________
City ______________________
State ____________ Zip________
Phone ______________________

OHIO SCIENTIFIC
1333 S. Chillicothe Road
Aurora, Ohio 44202
(216) 562-3101

01

The C3-S1

by Ohio Scientific

Possibly the world's most popular floppy disk based microcomputer.

Since its introduction in August, 1977, the Challenger III has gained *tremendous* acceptance in small business, educational and industrial development applications. Thousands of C3-S1's have been delivered and today hundreds of C3-S1 demonstrator units are set up at computer retailers around the country.

Why has the Challenger III become so successful in the fiercely competitive microcomputer industry? Here are just a few of the possible reasons.

- The Challenger III is the fastest microcomputer in BASIC (see "BASIC Timing Comparisons," *Kilobaud*, October, 1977, where Ohio Scientific out benchmarks all competitors).
- The Challenger III is the only computer system with a 6502A, 6800 and Z-80 offering the programmer all popular micros for maximum versatility.
- The C3 is backed by the largest base of systems level software for any microcomputer system including:
  - For the 6502A:
    - Microsoft 6 and 9 Digit BASIC
    - Assembler Editor
    - Word Processor
    - OS-65D Development DOS
    - OS-65U End User DOS with Extended BASIC
      - For Floppys
      - Winchester Hard Disks
      - Multi-users (Level 2)
      - Distributed Processing (Level 3)
  - For the 6800:
    - Floppy DOS
    - Assembler Editor
  - For the Z-80:
    - Floppy DOS
    - Microsoft Disk Extended BASIC
    - Microsoft FORTRAN
    - Microsoft COBOL
    - Macro Assembler and Editor
    - And Much More
- The C3 supports OS-65U, the ultra high performance "virtual data memory" DOS for floppys and hard disks which makes complex file structures like multi-key ISAM easy to use.
- The C3 is backed by a large library of applications programs and can make use of the tremendous amount of BASIC programs offered by independent suppliers and publishers because it uses Microsoft BASIC, the standard of the industry. Complete turnkey and custom business packages are available for the C3 from most OHIO SCIENTIFIC DEALERS.
- The C3 electronics and software are available in alternate mechanical configurations for special applications including the C3-OEM for volume users and the C3 letter series (C3-A, C3-B) which are optimized for use with hard disks.
- C3 systems are always delivered ready to use with 32K static RAM, dual floppys for 500K bytes of on-line storage and an RS-232 port strappable from 75 to 19,200 baud all *standard* in the minimum configuration.
- C3 systems offer the greatest expansion capability in the microcomputer industry. The C3 series supports OHIO SCIENTIFIC'S full line of over 40 expansion accessories. The maximum configuration is 768K bytes RAM, four 74 million byte Winchester hard disks (CD-74), 16 communications ports, real time clock, line printer, Word Processing printer and numerous control interfaces.
- C3 systems have phenomenal performance-to-cost ratios. The C3-S1 base price with 32K RAM, dual floppys, RS-232 port complete with 8K BASIC and DOS is under $3600 and expansion accessories are comparably priced. For example, the CD-74, 74 million byte Winchester disk complete with interface and OS-65U operating system at about $6000.

The C3 series is quite possibly so successful because it offers the highest hardware performance, best software support, most versatility and greatest expandability in the microcomputer systems market at nearly the lowest price in the industry.

For more information, contact your local OHIO SCIENTIFIC DEALER or the factory at (216) 562-3101.

OHIO SCIENTIFIC

1333 S. Chillicothe Road • Aurora, Ohio 44202

PUBLISHER'S REMARKS

Wayne Green

The Challenger IP—outside.

A Visit To Ohio

Scientific, that is.

With the slacking off of computer shows in July (only one, in Virginia, near Washington), Sherry and I had a chance to take a day off and zip out to Ohio Scientific in Aurora, not very far from Cleveland.

It was prophetic that we should pass the Computer Shop display of an OSI system in the airline terminal on our way to visit OSI. I grabbed one of the OSI brochures, which were available at the little kiosk, so I could read up on the OSI products during our trip.

The Ohio Scientific building is impressive . . . over 50,000 feet of work space . . . something to make me jealous. OSI's new building houses most of the production and test facilities. They still are doing much of their product development and software work in the old plant in nearby Hiram, Ohio.

While some in the industry are concentrating on fighting the low-end price battle, and others are on the high end, OSI is trying to cover all bases.

Their new Challenger IP weighs in at $279 and provides a remarkable amount of computing for this incredible price. The thought, obviously, is to provide an alternative to the TRS-80 for computer stores to sell.

I think the first public showing was at the Philly show in August, where they had it on demonstration. This new system is even more amazing than the Challenger IIP, announced last year in response to the TRS-80. When you take the system out of the cabinet you see how simple and well designed it is.

With the IP system, OSI has introduced a strong price competitor for the KIM, but with a full keyboard and output for a video monitor . . . complete with BASIC in ROM! When a test unit arrived at my office, I had it running with a program in about ten minutes. Oh, I had some trouble loading the cassette at first, but a change of recorders fixed that. Some programs would load and some wouldn't . . . probably due to the head of my first recorder not being cleaned recently.

On the high price end is the C3B system. The photo shows a top-of-the-line C3B system used at OSI for software development. This outfit, which includes the 74 MB disk, two floppy disks, a processor that has three microcomputer chips—6502, 6800, Z-80—so you can use programs developed for any of the popular microcomputer systems, a CRT terminal and the OSI operating system, sells for under $13,000. If I didn't already have about $40,000 invested in programs for our Prime system I'd jump for the C3B in a minute. That would give me a lot more computing for about one-fifth the price.

The IP and C3B are put together in OSI's assembly area, which keeps a lot of people very busy. It's a b-i-g place. On one side of the assembly area I came across a C3-based word-processing system, complete with Diablo Hy Type printer.

Business Articles—and Beyond

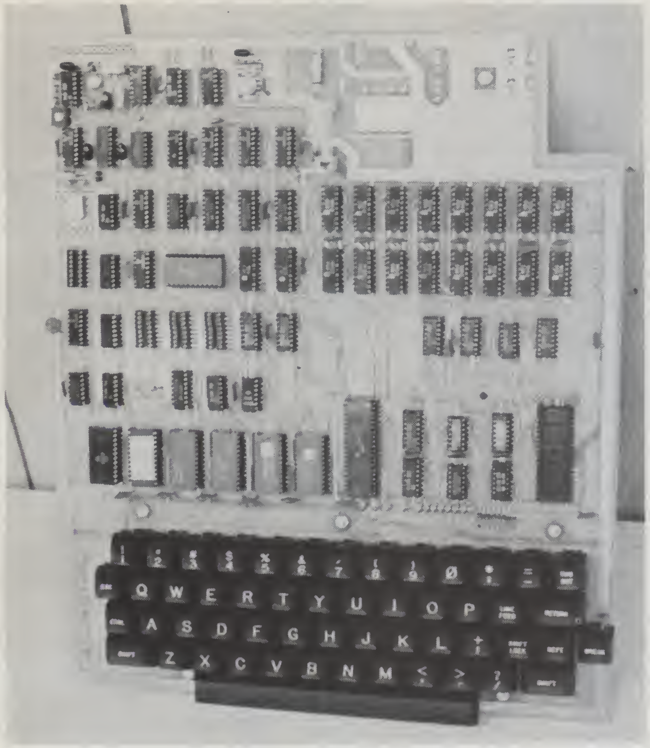
Obviously many OSI systems are finding their way into business applications. On the off chance that I may not have made my intentions clear, I reiterate that I'm looking for articles for *Kilobaud MICROCOMPUTING* on how microcomputers are doing in business. I'd like to have the articles written in plain English, with as few buzzwords as possible. The articles should tell us what system was used, what operating system, where the programs were obtained or how they were developed, what problems arose and how they were surmounted, how the system works at present and what expansions are contemplated.

The OSI C3 system seems ideally suited for school use, where it can not only help the school keep track of all of its students and their grades, but could also act as a data base for storing educational programs that could be used via smart terminals by

Computer Shop display.

OSI's building.

OSI's large assembly area.



The IP—inside.

students. If anyone runs into some applications such as this, *Kilobaud MICROCOMPUTING* is most interested in an article.

Programmers might also keep in mind that any programs developed for use on business systems could possibly be of value for more widespread distribution via a program publisher such as Instant Software, Inc. Programs will be selling for whatever seems a reasonable price, with some being projected to run as high as $800. At that rate the royalty on a sale through computer stores would run to about $8.5 million for every 100,000 sold.

Why Most Programs Are Worthless

Several of the people working on Instant Software have been surprised at the third-rate quality of many programs being submitted for publication. I've heard the same complaint from most of the other people who are getting programs for possible publication too. How come?

If you mull over the history of software for microcomputers, the picture should come into focus and the quality of most of the programs written so far should be understandable.

Although it's true that a few businessmen have gone out and bought microcomputers, and then gone to the trouble to personally write the programs needed to use their systems, most microcomputer users have had less serious plans for their systems and have made do with what they could scrounge from magazine articles or books of programs, or borrow from friends.

The handful of dedicated programmers who have bought systems seem to have written debuggers, disassemblers, utilities and a rash of forgettable games. While I'm sure that the utility programs are great to have, the demand for them has been relatively small, and, as far as I know, no one has made anything more than egg money from such efforts.

So along comes Instant Software and a call for original programs—and what do we get? A lot of forgettable games. Fortunately there are a few diamonds among the rubble . . . and just a hint of the programs to come for business, for teaching and for practical uses of our microcomputers.

So far, most of the businessmen who have patiently worked out programs for their own use have not really become aware of the gold mines they have generated. Once they perceive the value of their work, they will submit it to the bigger program publishers. The only danger in this is that there is a finite need for each specific type of program, and if someone else gets there first, *he* gets the bonanza.

When the plan for Instant Software was announced, a few sharp programmers wasted no time in submitting programs . . . and they are in line for the payoff. With the dramatic need for hundreds—even thousands—more programs, it is still very early in the game. In a few months it is going to be a whole lot tougher to sell a program to a publisher because the large amounts of money involved are bound to bring in heavyweight programmers and even systems houses.

Consider that a first shipment of programs, even if it goes to only about half of the computer stores, say 500 stores, with ten copies per store, comes to 5000 copies for the initial sale. That would bring in well over $4000 in royalties from the first run. We plan to be the major supplier of programs for Radio Shack, and that could bring the initial run up

C3B System in operation.

C3-based word-processing system.

kilobaud MICROCOMPUTING

Publisher/Editor
Wayne Green
Executive Vice President
Sherry Smythe
Assistant Publisher/Editor
Jeffrey D. DeTray
Managing Editor
John Barry
Editorial Assistants
Dennis Brisson
Susan Gross
Administrative Assistant
Dotty Gibson
Production Department
Manager:
Lynn Panciera-Fraser
Staff:
Craig Brown
Gayle Cabana
Robert Drew
James H. Gray II
Bruce Hedin
Carl Jackson
Dion Owens
Noel R. Self
Robin M. Sloan
Typesetting
Barbara J. Latti
Sara Bedell
Mary Kinzel
Holly Walsh
Photography
Bill Heydolph
Tedd Cluff
Associate Editors
John Arnold
Phil Feldman
Steve Fuller
Rod Hallen
Tom Rugg
Peter Stark
Sherman P. Wantz
Dick Whipple
Ken Widelitz
Bookkeeper
Knud E. M. Keller
Marketing/Circulation Mgr.
Robert LaPointe
Bulk Sales Mgr.
Judy Waterman
Asst. Circulation Mgr.
Mary Beth Stanek
Circulation
Pauline Johnstone
Customer Service
Joyce Tarr
Computer Programming
Richard Dykema
Instant Software Micro Lab
William F. Gollan
Alan Anderson
Walter Asonevich
Shirley Mahoney
Scott Spangenberg
Cida Teixeira
Palmer T. Wolf
Stan Wszola
Advertising
Marcia Stone
Nancy Ciampa
Rita Rivard
Pat Theriault
Bill York
European Marketing Dir.
Reinhard Nedela
Australian Distributor
Katherine Thirkell

Cover: The new MSI System 10 from Midwest Scientific Instruments, Olathe KS.

Kilobaud MICROCOMPUTING is published monthly by 1001001, Inc., Peterborough NH 03458. Subscription rates in U.S. and Canada are $18 for one year and $45 for three years. In Europe Kilobaud erscheint montatlich bei Fachzeitschriftenvertrieb Monika Nedela, 7778 Markdorf, Markstr. 3. Australia: For subscriptions write—Katherine Thirkell, Sontron Instruments, 17 Arawatta St., Carnegie, Vic 3163 Australia. Please write for other foreign rates. Second-class postage paid at Peterborough NH 03458 and at additional mailing offices. Publication No. 346690. Phone 603-924-3873.

INCLUDE OLD ADDRESS AND ZIP CODE WITH ADDRESS CHANGE NOTIFICATION

another five to ten thousand, at least. If Radio Shack expands TRS-80 sales to their entire fleet of nearly 10,000 stores, the program needs will be substantial.

It takes a large organization to bring about those sales figures. We're estimating the first-year costs of running Instant Software at above $1 million for salaries alone, and that's just for publishing and distributing programs. If you horse around with a small outfit you'll probably be selling your programs for peanuts.

What Is Needed?

An insatiable need exists in the business field for programs of both general and specific nature. For instance, printers need to be able to estimate printing costs, which would be simple to computerize. They need a program for figuring page impositions. You'll do better to start writing programs for a field with which you are intimately familiar and work into unfamiliar fields later, getting help from friends who are in those fields.

Relatively small programs will be selling for the usual $7.95. The bigger, more-comprehensive ones will be selling for higher figures. I expect that complete sets of programs for a specific business—programs that definitely will help a dealer sell a system that runs perhaps $12,000—will easily sell for $1000 (with a nice royalty, obviously). We'll try to do the best we can on program pricing, keeping in mind supply and demand.

Educational programs are just barely starting. Those submitted so far have not been outstanding . . . with but a few exceptions. You know that a program which teaches BASIC programming is going to sell well in this field. I suspect that programs which teach the fundamentals of any other language will also sell briskly. Then might come programs for teaching the basics of computer electronics, such as gates and flip-flops. Eventually I expect we will carry a catalog of several thousand educational programs.

I expect that Instant Software will be pioneering in the development of video/computer programs. This will be a marriage of television and microcomputing. The early systems will probably be made up of a video recorder and computer, with a box between to do the work. Later we'll be seeing video teaching systems with the computer built in, and possibly even including a small TV camera so the student will be able to communicate via cable with a teacher.

How far off is the day when an educational program is budgeted at $500,000 or more? Not that far, when you consider that such a program could be used to teach several million people . . . and used for years. We may even see budgets of over $1 million for single teaching programs, and they would still turn a profit.

For the time being, let's get cracking on some first-rate programs to get businesses interested in buying microcomputers. You

(continued on page 21)

Reader Responsibility

One of your responsibilites, as a reader of *Kilobaud MICROCOMPUTING*, is to aid and abet the increasing of circulation and advertising, both of which will bring you the same benefit: a larger and even better magazine. You can help by encouraging your friends to subscribe to *Kilobaud MICROCOMPUTING*. Remember: Subscriptions are guaranteed—money back if not delighted, so no one can lose. You can also help by tearing out one of the cards just inside the back cover and circling replies you'd like to see: catalogs, spec sheets, etc. Advertisers put a lot of trust in reader requests for information. To make it more worth your while to send in the card, a drawing will be held each month and the winner will get a lifetime subscription to *Kilobaud MICROCOMPUTING*!

This month's winner of a lifetime subscription to *Kilobaud MICROCOMPUTING* is Ray Slattery of Saddle Brook NJ.

OUTPUT FROM ISI

Sherry Smythe

The purpose of this column is to provide some inside information on what has been happening in software publishing . . . and what is needed.

Now that Instant Software, Inc. (ISI) has moved into its new office building there is finally room for more employees. We have a new marketing manager, whose job it will be to get ISI products into every computer store in the world. The new project coordinator is Bill Gollan. He keeps things moving and coordinates the three main departments: program editing, production and marketing.

The new microcomputer lab, with its 30 systems, is certainly impressive. This allows the staff to check submitted programs quickly for just about any popular system. It's amazing how much faster things have been going now that everyone has room to work. Look for at least 20 new programs to be announced in the next issue of *Kilobaud MICROCOMPUTING.*

We've been receiving requests from dealers for Apple programs, but we haven't been getting many Apple programs. Don't miss the boat on having your Apple programs published if you've written anything significant. Remember that the first-come of each type of program will be the big winner on royalties. Coming in second doesn't pay off in programming any more than it does elsewhere.

We need good game programs for all systems—programs that will play a tough game of backgammon, chess, checkers, cribbage, etc. They have to be able to beat good players.

Good educational programs are also important. The latest Radio Shack ad campaign is aimed at students (or at least at the parents of students). This means there is going to be a whale of a market for educational programs. They should have good graphics and be fun to use so kids will *want* to learn.

If you're interested in making money at home while having fun with your microcomputer, what better way is there than writing programs and getting royalties for them. Your equipment and office constitute business expenses, and those royalty checks can quickly mount up to more than many regular salaries. This can give you a freedom beyond most people's dreams. You might want to put your computer in a van and travel, writing programs as you go.

ISI is also involved in an investment opportunity for small (or large) investors with guaranteed returns of up to 15 percent per annum. Programmers interested in this type of money-making venture should contact me directly for more information. We're talking about cash outlays of from $250 and up for one to five years' duration.

ISI Policies

Since ISI now has its own building, submissions of articles for *Kilobaud MICROCOMPUTING* should be kept separate from ISI submissions. If you do intend a program for both *KM* and ISI, clearly state this in your cover letter, and don't forget the stamped self-addressed return envelope.

We insist on having a signed contract before we give a program a preliminary review, and several programmers have wondered why. The main purpose of the contract is a guarantee that the program is original and not a thinly disguised Hammurabi or other old standard. This also weeds out the user-group officials who try to sell donated programs to us . . . which has happened. We really can't afford to spend a lot of money processing and reviewing a stolen program that we can't publish.

We have other problems to contend with. For example, our associate-editor plan has been held up by a bug in the computer program for keeping track of associate editors. Such a business!

If you have any questions about possible programs, write. Please don't call, as this brings everything to a halt, often at a key moment. We haven't yet found anyone whose exclusive job it will be to sit and answer questions. The job is open. . . ?

PET-POURRI

Len Lindsay

Last month I listed some accessories for the PET, including printers, floppy disks, keyboards and extra memory. Of course, there are many more. Microsignal (Box 161988, Sacto CA 95816) is marketing a voice-input module called Compuvox for $29.95. With this unit your PET can recognize sound and no sound. It cannot distinguish between different words. Thus, "one" and "two" are equivalent; but "go down" and "up" *are* different. "Go down" is two sounds and "up" is one sound. So to have PET recognize four commands you might use: "up," "go down," "now turn right" and "go to your left."

Microsignal also markets a sound-output unit (a speaker) called Computone for $14.95. With a speaker attached to your PET you can add sound effects to your programs and even have the PET play songs (i.e., the theme from *Star Wars* is available from the *PET Gazette*). Computone is compatible with the proposed sound conventions, which are explained in this column. The Music Box, sold by New England Electronics (248 Bridge St., Springfield MA 01103) for $49.95, is not compatible with the sound conventions, but is very professionally packaged and comes with programs allowing you to actually compose music with your PET.

Joysticks that just plug into your PET are available from Coyote Electronics (PO Box 101, Coyote CA 95013) . . . $50 for a complete package deal. A beautiful connector box, which has no bare wires and appears to be safe for use around children, plugs into your user port. The joysticks then simply plug into this connection box. It is very well designed. Along with the box you get two Atari joysticks, a test program, two game tapes and a very complete three-part instruction series all on cassette, ready to load and go. Clear and detailed programming instructions make it easy to add joystick control to all your games.

A digital plotter for the PET is available for $195 from X and Y Enterprises (PO Box 796, Huntsville AL 35804) in kit form ($249 assembled). A light pen for the PET costs only $24.95 from 3 G Company (Rte. 3, Box 28A, Gaston OR 97119). Your PET can then recognize which part of the screen you touch with the light pen.

If you wish you could use some of the S-100 boards, you will be interested in Betsi, an S-100 adapter for the PET. It is made by Forethought Products (87070 Dukhobor Rd., Eugene OR 97402). You may remember Forethought's Kimsi S-100 adapter for the KIM. Betsi costs $119 as a kit or $165 assembled.

Add Sound to Your PET

With a speaker/amplifier, all you need is two wires to add sound to your PET. Here is how to do it:

1. Get an edge connector for the user port (the center connector in back of your PET).
2. Get a speaker/amplifier (amplifier is a must). Radio Shack has one for $10.95 in a small plastic case with a plug-in jack for your cord. For simplicity's sake also get their cord ($1.95), which plugs into their speaker/amplifier and has two bare wires on the other end.
3. Of the two wires coming into the PET from your amplifier, one is the ground, the other carries the data for the sound. The two wires hook to pins M and N (the two pins on the bottom right). Ground is hooked to pin N (the last pin on the bottom), and the other wire is hooked to pin M (next to pin N). Solder them onto your edge connector. Don't solder onto the PET!
4. With your PET turned off, plug in the connector. Be sure it also is plugged into the amplifier.
5. Now turn on your PET and amplifier, and you're off.

Sound Programming

You have three POKE locations that together produce sound.

POKE 59467,16 turns on the sound mode (and turns off your ability to use your tape units).

POKE 59464,X plays a note determined by X (X can be from 1-255; 1 is highest, 255 is lowest).

POKE 59466,Y allows you to change the octave (three octaves available, Y = 15, 51, 85. Y can be any integer from 1 to 255).

Each note stays on until another note begins or it is turned off. At the end of every sound subroutine, make sure to POKE a 0 in each location (to regain tape functions). Try this example:

```
3000 POKE59467,16:REM sound on
3010 POKE59466,51
3020 FOR P=1 TO 255
3030 :POKE59464,P:REM play the note
3040 NEXT P
3050 POKE59467,0: POKE59466,0:
     POKE59464,0
3060 REM sound off/tape functional
```

Software

As you know, the PET comes with Microsoft BASIC in ROM. Now you can use other high-level languages with your PET.

PILOT is an easy language to learn and use. Invented by John Starkweather, it is most useful in teaching children about programming. Programming is easy since there only are five commands—T: (type), A: (ask), M: (match), J: (jump) and C: (compute). The C: command allows you to use most BASIC commands if you wish. It is available from the Peninsula School Computer Project, Peninsula School, Peninsula Way, Menlo Park CA 94025. For only $19.95 you get the PILOT interpreter on cassette along with five sample game programs written in PILOT. The 24-page manual includes program listings and tips on how to use PILOT.

FORTH is a language now frequently discussed. Fortunately there is a version available for the PET, appropriately called PETFORTH. It can be obtained from Programma Consultants, 3400 Wilshire Blvd., Los Angeles CA 90010, for $35. FORTH is a structured language that can run faster than its equivalent assembly code might. PETFORTH works and will be supported.

Simulations are often the most exciting and realistic of the computer games. Personal Software (PO Box 136-L, Cambridge MA 02138) has several well done simulations for the PET. Kingdom is one of my favorites. It is a big improvement over the varieties of Hammurabi I've played. You are the ruler of a kingdom and must make decisions such as how much to feed your people, how much grain to plant and how much land to buy or sell. Background information needed to make your decisions is given at the beginning. The game is realistic, with rats eating part of your grain and attacks by the Huns, killing some of your people. Kingdom, Poker, Matador and One Queen are on one cassette for $14.95. Another cassette, titled Stimulating Simulations, is also $14.95. It includes a 64-page instruction book and ten simulation games.

Mike Richter has developed a very sophisticated simulation series. He refers to the games included as Hypergames. In these, you and your party of Goodguys must find your way through a computer-generated "world" in search of treasure. Badguys are also in this "world" and try to stop you. You are affected by your domain as well as by the Badguys. Most ingenious about Mike's first Hypergame is that the program is *all* variables. A specific game is created with a data tape that defines who the Goodguys and Badguys are, how they affect each other and the layout of the "world," along with its effects on you.

Firstworld is Mike's first specific Hypergame. You and a Fighter, Burglar, Witch and Priest are the Goodguys. You meet the Dragon, a Swarm of Gnats and an Ogre along your way in search of the treasure. Each encounter is a battle. Some of your party may be injured. Fortunately, injuries heal as time passes, so with luck you will make it through the Rushing River, Waterfall, Kansas Cyclone and other treacherous areas of Firstworld. But then you must find the Exit—which is not the same place that you came in. A wonderful game.

Mike also has a program called Hyperwriter, which you use to create data tapes for specific games. That data could then be used with any Hypergame (several different Hypergame programs are being devised). With it Mike created Firstworld, and I have created a specific game, Test Island, which works beautifully with Hypergame 1.

The series is still being documented and human-engineered, so it is not yet being marketed. Further information will be in next month's column, or feel free to contact Mike at 2600 Colby Ave., Los Angeles CA 90064.

Publications

Greg Yob is writing *The PET Manual*, which will be published by Mind's Eye Software (PO Box 354, Palo Alto CA 94301). This

comprehensive manual should be available February 1979 for about $16. (*Kilobaud MICROCOMPUTING* plans to preview part of *The PET Manual.*)

Another PET manual is being marketed by PETABLE, whose ads say: "Ever look at Radio Shack's 232 pg. owner's manual and wish Commodore had one? We have just published *PETABLE*, an introduction to PET BASIC, graphics and capabilities. It includes programs, exercises, software/hardware sources, a periodical index and much more. Send check or money order for $4.95 to: PETABLE, PO Box 461, Philipsburg PA 16866."

It is implied that this manual is as good as Radio Shack's (an excellent manual). *PETABLE* is 41 pages (22 sheets of 8 1/2 by 11 paper stapled together) and does *not* even have a table of contents. Some pages have parts of words going right off the paper's edge. Furthermore, since the booklet is stapled down the left side, many words are hidden in the folds and not readable. *PETABLE's* software/hardware sources consist of merely listing Newman Computer Exchange and advising you to write for their catalog of PET products.

PETABLE contains many errors and the material is poorly presented. Important items are mentioned, but no information is given. It just says, "Try it and see what happens." For example, page 7 reads, "Take a look at the Boolean operatives AND, NOT, OR on page 45 of your PET manual and try using them with string variables." That is a complete section! It is representative of *PETABLE*. This "manual" will probably confuse rather than help. I recommend avoiding it.

An excellent source of technical information on the PET is the *PET Newsletter* (Sphinx). Six issues cost $4.50 and can be ordered from: PET Newsletter/Computer Project, Lawrence Hall of Science, University of California, Berkeley CA 94720. Make checks payable to: Regents of the Univ. of Calif. This newsletter has contained memory maps, machine-language programs, explanations of various PET functions and I/O information. Highly recommended.

If you hope to do any machine-language programming, you will be interested in these books: *Programming a Microcomputer: 6502* by Caxton Foster, $8.95, published by Addison-Wesley, Reading MA 01867, and designed especially for KIM, PET and Apple users. You will also want the two manuals from MOS Technology: *6500 Programming Manual* and *6500 Hardware Manual*. Prices vary. AB Computers (PO Box 104, Perkasie PA 18944) sells them for $6.50 each.

```
3 REM******************************
4 REM***** ON GOSUB EXAMPLE *****
5 REM******************************
6 REM BY L LINDSAY
7 REM*******************************
8 REM***** ASSIGN POKE VALUES *****
9 REM*******************************
10 BALL=81
20 TRAIL=87
30 UPWALL=98
32 BOTTUMWALL=226
34 LEFTWALL=225
36 RIGHTWALL=97
40 SCREENSTART=32768+500
50 POKE59468,12:REM GRAPHICS MODE
60 PRINT"♥":REM TYPE 60 ?"[CLR]"
97 REM*************************
98 REM***** DRAW BORDERS *****
99 REM*************************
100 FOR I=1 TO 25
110 :POKE 32767+(I*40),RIGHTWALL
120 :POKE 32767+(I*40)+1,LEFTWALL
130 NEXTI
140 FOR I=1 TO 40
150 :POKE 32767+I,UPWALL
160 :POKE 32767+960+I,BOTTUMWALL
170 NEXT I
197 REM*********************************
198 REM**** ILLUSTRATE DIRECTIONS ****
199 REM*********************************
200 REM
210 REM   7  8  9
220 REM    \ ! /
230 REM     \!/
240 REM  4---5---6
250 REM     /!\
260 REM    / ! \
270 REM   1  2  3
280 REM
297 REM*************************
298 REM***** GET DIRECTION *****
299 REM*************************
300 POKE525,0:WAIT525,1:GET DIRECTIN$
310 DIRECTIN=VAL(DIRECTIN$)
320 IF DIRECTIN=0 THEN GOTO 300
397 REM****************************
398 REM***** ON GOSUB SECTION *****
399 REM****************************
400 ON DIRECTIN GOSUB 1000,2000,3000,4000,5000,6000,7000,8000,9000
500 POKE SCREENOLD,TRAIL
600 Q=PEEK(SCREENOLD+ADJUST)
610 IF Q=UP OR Q=BOTTUM OR Q=RIGH OR Q=LEFT THEN GOTO 710
700 SCREENNOW=SCREENOLD+ADJUST
710 POKE SCREENNOW,BALL
720 GOTO300:REM GET NEXT DIRECTION
996 END:REM SUBROUTINES FOLLOW
997 REM**************************
998 REM***** GOSUB ROUTINES *****
999 REM**************************
1000 ADJUST= 39:RETURN:REM DIRECTION IS 1
2000 ADJUST= 40:RETURN:REM DIRECTION IS 2
3000 ADJUST= 41:RETURN:REM DIRECTION IS 3
4000 ADJUST= -1:RETURN:REM DIRECTION IS 4
5000 ADJUST=  0:RETURN:REM DIRECTION IS 5
6000 ADJUST=  1:RETURN:REM DIRECTION IS 6
7000 ADJUST=-41:RETURN:REM DIRECTION IS 7
8000 ADJUST=-40:RETURN:REM DIRECTION IS 8
9000 ADJUST=-39:RETURN:REM DIRECTION IS 9
READY.
```

Example program.

Programming Tips

Last month we discussed, among other things, the GET command. This month's example program uses the GET command differently in line 300. Location 525 tells your PET how many keys have been pressed and are stored in the keyboard buffer (locations 527-536). In line 300 we first POKE a 0 in the keyboard buffer counter so the PET thinks no keys have been pressed. Then we use the WAIT command to tell our PET to just WAIT until its keyboard counter says a key has been pressed and stored in the keyboard buffer. It will then continue executing the program. The next statement tells it to GET a character (we are sure there is one character in the keyboard buffer now). Thus a sophisticated GET routine is:

```
POKE 525,0 :WAIT 525,1 :GET A$
```

You then have the option of printing that character or not.

This month the Example program illustrates the ON . . . GOSUB commands. The program narrative follows.

In PET BASIC, variables can be up to 255 characters long. Thus you can name variables with recognizable titles, making reading the program listing easier. You must avoid using variables that contain any of the reserved BASIC commands such as ON, TO, OR, AND and LET. This applies to such commands as:

```
IF Q = RIGHT OR Q = LEFT THEN GOTO
   710
```

Your PET does not like that line because RIGHT OR includes the TO command (PET ignores spaces so RIGHTOR contains TO). Thus BOTTOM becomes BOTTUM (to avoid the TO) and DIRECTION becomes DIRECTIN (to avoid the ON).

This program will allow you to move a ball around the screen using your numeric keypad for directions. There are no PRINT commands other than line 60, which clears the screen. Your PET uses memory locations 32768 to 33767 for the screen display. Thus by POKEing into those locations you can write on your screen. There are 25 lines of 40 characters (1000 total).

First we assign to the variables the correct numbers to be used in the program with POKE statements. Also note that in line 40 we identify the center of the screen. PET accepts the variable SCREENSTART but will remember its value as being SC. (The

(continued on page 21)

YOU ASKED FOR IT
YOU GOT IT
DSI
QUIK-KIT®

550 MHZ COUNTER KIT

Performance You Can Count On

OPERATES ON

- Batt 6-C Size
- DC 8.2 To 14.5 VDC
- AC Batt. Eliminator

$99.95

MODEL 3550 KIT

DSI OFFERS THE BEST OF TWO WORLDS . . .

An unprecendented DSI VALUE . . . in a high quality, LSI Design, 550 MHZ frequency counter kit. And, because it's a DSI innovation, you know it obsoletes any competitive makes, both in price & performance. The basic 550 MHZ counter & time base are factory assembled, tested and burned-in. The problems of bad LEDS, IC's, capacitors, are a thing of the past with DSI QUIK-KIT®. But you can take pride in assembling the power supply, PC mounted selector switch, input connectors, and the final mechanical assembly of your 550 MHZ counter, into its' handsome cabinet. **GO WITH THE LEADER . . . BUY A DSI FREQUENCY COUNTER KIT. SAVE TIME & MONEY AND BE ASSURED IT WILL WORK THE FIRST TIME.**

SEE YOUR LOCAL DEALER
OR
CALL TOLL FREE (800) 854-2049
Californa Residents, Call Collect (714) 565-8402
DSI INSTRUMENTS, INC.
7914 Ronson Road No. G, San Diego, CA 92111

VISA • MC
AMERICAN EXPRESS
CK • MONEY ORDER
COD

SPECIFICATIONS

Time Base TCXO 1PPM 65° to 85°F
Frequency Range 50HZ to 550MHZ
Resolution 1HZ to 55MHZ, 10HZ to 550MHZ
Gate Time 1 second - 1/10 second
Sensitivity 25MV 150 & 250MHZ 75MV 550MHZ
Display Eight 1/2-inch LEDS
Input Two SO239 Connectors
Power 6C-Size Batt., 15HR, or 8.2VDC to 14.5VDC
Current 150 Ma standby 300 Ma operational

3550 KIT INCLUDES

- Pre-assembled, tested counter board
- Case, power supply, connectors, hardware
- Built-in prescaler & preamp
- Gate Light - Automatic Zero Blanking
- Automatic Decimal Point
- One to two hours assembly time
- One Year Warranty on all parts
- All new parts - not factory seconds or surplus

3550 Kit	**$99.95**
T-101 Telescopic Antenna	**3.95**
AC-9 Battery Eliminator	**7.95**
Cigarette Lighter DC Adapter	**2.95**

TERMS: Orders to U.S. and Canada, add 5% to maximum of $10.00 per order for shipping, handling and insurance. To all other countries, add 15% of total order. California Residents add 6% State Sales Tax.

NEW PRODUCTS

Edited by Dennis Brisson

On-board EPROM Capability

Electronic Product Associates, Inc., 1157 Vega St., San Diego CA 92110, is offering their Micro-68 Computer System with on-board EPROM capability. The Micro-68 is a low-cost, complete, ready-to-use microprocessor offering an economical solution for both scientific applications and industrial usage.

Built around the Motorola/AMI/Hitachi 6800 microprocessor, the Micro-68 comes with its own integral power supply, 16 button hexadecimal keyboard, six-digit LED display, 128 words of RAM (expandable up to 768 words) and 2K of user programmable EPROM. The 512-word MON-1 bus PROM contains all the service routines necessary to load programs easily, inspect and edit them as necessary, insert breakpoints for debugging and execute programs. It is ideal for decreasing the learning curve and prototyping.

Studio II

Radio Shack has been selling a video game called Studio II, which offers computer hobbyists a potential bonanza. It seems Radio Shack bought up a large stock of these RCA units just in time for the holiday season. What do you get for your $59.95? Well, it turns out that the heart of the Studio II is RCA's 1802 microprocessor chip. What's more, we've been promised an article showing how to convert the Studio II into a true-blue micro with very little hassle. Better grab one while they last.—Editors.

SWTP Intelligent Terminal

Southwest Technical Products announces the CT-82 Intelligent Terminal. It will work with almost any modem or computer system interfaced RS-232 serially from 50 through 38,400 baud.

The CT-82 features over 100 control functions operable from either the CT-82's keyboard or the computer's program. With its low price and graphics capability, it's ideal for business or hobby use.

The unique terminal design utilizes a Motorola 6802 microprocessor and 6845 CRT controller integrated into a modular system that is simple, reliable and easy to service. Other features include 7 × 12 matrix, upper and lowercase letters, 82 × 16 or 92 × 22 format—plus graphics and a printer output port. The CT-82 is offered in assembled form only and sells for $795.

SWTP, 219 W. Rhapsody, San Antonio TX 78216.

S-100 Display-Sense Board

Digital Dynamics has recently come out with a new S-100 Display-Sense Board that adds basic front panel features to front panel-less computers. It can be used to augment mainframes already having front panels by providing an additional Sense Switch port and on-card bus displays for troubleshooting. The board provides visual readout of Address lines, Data lines and six of the more commonly used Status lines. Optional IC sections and connector pads allow user access to most of the remaining S-100 Status and Control bus lines.

The Sense Switch feature allows you to input data words to the CPU via an I/O port. The Sense Switch may be addressed at the conventional 377 octal/FF hex or may be relocated to any of the remaining 255 locations by on-board address jumpers.

The Mode and Sense switches are remotable to the front panel by adding a 16-pin DIP plug and ribbon cable. The Display-Sense Board is ideal for hardware and software debugging and as a maintenance and programming aid. Cost is $89.95.

Digital Dynamics, Inc., PO Box 27243, San Antonio TX 78227.

The Accountant.

CDS Business System

The Accountant is a complete turn-key system with a Versatile 4 Dual Drive computer, an Impact Printer, application business software and a Formica table on roll-away casters for convenient moving.

The business software includes General Ledger, which will set up a chart of accounts and general journal and will produce a trial balance, income statement and balance sheet. It will automatically post transactions to the Ledger and produce an audit trail of transactions. Accounts Payable programs record statements by vendor, date or through a range of dates and make cash projections. They will also print checks and mailing stubs and journalize transactions so they will be processed by the Microledger. Accounts Receivable programs produce statements by customer, date or range of dates as well as aged accounts receivable report. It will report cash projections from collections and journalize transactions for processing in the Microledger.

The Accountant also includes Inventory programs and Personnel/Payroll programs. The entire package costs $6000.

Computer Data Systems, Inc., 5460 Fairmont Drive, Wilmington DE 19808.

The Micro-68 from EPA.

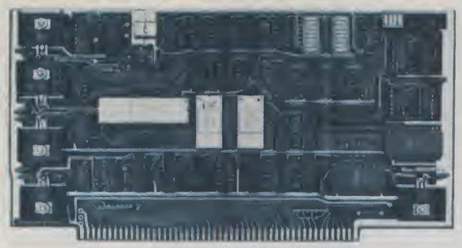
The CT-82.

CPU and Video Boards

SSM (formerly Solid State Music) announces two new boards for S-100 computers: the CB1 8080 CPU board and the VB2, an I/O controlled video interface board.

The CB1 contains enough RAM (256 bytes), EPROM (2K of 2708) and other features to allow a two-board computer; all you need is an I/O or video board.

For operation without a front panel, the CB1 can *vector-jump* to the beginning address of the on-board EPROM on power-up or reset . . . and the board can



CB1 8080 CPU Board.

VB2 Video Board.

generate an MWRITE signal. The jùmp circuit and MWRITE signal can be disabled if necessary. An 8-bit parallel input port with separate status is provided on the CB1, with DIP-switch addressing up to 31 decimal. The CB1 is available in kit or assembled form; kit price is $144.95.

The VB2 has its own keyboard input port, so there is no need for another I/O board for either keyboard or video display. The hardware-controlled cursor for line feed, carriage return, back space and clear-screen frees up valuable memory space.

The display is 64 × 16 with uppercase letters and is switch-selectable for white-on-black or black-on-white. The board features full interlace for complete compatibility with U.S. TV video standards; circuitry is provided to drive an external speaker for "beep" tone. The VB2 is available in kit ($149.95) or assembled form.

SSM, 2116 Walsh Avenue, Santa Clara CA 95050.

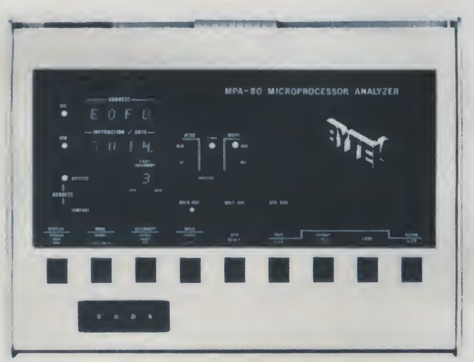
Microprocessor Analyzer

The Model MPA-80 Microprocessor Analyzer is simple but effective in solving hardware or software problems in 8080- or 8085-based microprocessor systems. It provides real-time examination of the processor's operation and can single-step the program by instruction or machine cycle. Hardware breakpoint allows looping on instruction, memory or I/O.

The Reverse Trace mode can find how the program got where it is, as well as who called a subroutine or accessed a memory or I/O location. Hexadecimal displays provide optimum information for each type of instruction (op code, one or two data bytes, memory address or the actual branch address in the case of conditional JMPs, CALLs and RETs). Processor Status Indicators are extremely helpful.

Interrupts are displayed as they occur and may be disabled as well as DMA to facilitate detailed analysis. Personality modules (8080 or 8085) provide the interface to your system. Applications for the Model MPA-80 are in engineering, production, test and field service. Prices are $845 for the MPA-80 and $145 for the 8080 or 8085 Personality Module.

Bytek, Box 3026, Burbank CA 91504.



The MPA-80.

PERK Up Your PET

PERK, the Professional Encoded Remote Keyboard for the Commodore PET, is a plug-in, typewriter style, alphanumeric keyboard designed to enhance the operation of the PET. The nonstandard, block layout of the PET keyboard, only half the size of standard typewriter keys, makes touch-typing virtually impossible. The PERK standard keyboard, however, makes data entry convenient. It shares the PET internal keyboard interface, allowing the two keyboards to be used interchangeably. Both are active at all times, allowing the operator to use the PERK keyboard for normal data entry and the PET keyboard for numerics or graphic capabilities.

Housed in a custom steel desktop enclosure, the PERK is connected to the PET by means of a plug-in interface card. Easy installation requires no tools, soldering or assembly. Once installed, the PERK remote keyboard is immediately usable on all existing software. No changes or modifications are required, and both internal and external keyboards may be used simultaneously. A plug-in, UL listed power supply activates the PERK keyboard and interface, eliminating any drain on the PET power supply and protecting the user's warranty.

In addition to standard upper and lowercase alpha characters, with proper shifting the PERK keyboard "alpha-lock" allows easy entry of uppercase-only or "TTY" mode operation. Standard CRT terminal control functions are implemented for cursor control, and full screen editing capabilities are provided. More than one PERK keyboard may be attached to a single PET computer for multiple operator data entry. Wired and tested, the unit costs $229.95.

George Risk Industries, Inc., GRI Plaza, Kimball NB 69145.

PERK Keyboard plugged into PET.

TBEEP

Just as larger business computers come with beepers to save time by signaling the computer operator that an error has occurred or that some additional action must be taken to continue processing, TRS-80 business systems can now be equipped with a low-cost, easy-to-install and easy-to-use software-controlled beeper.

TBEEP produces a clear, distinct tone similar to that of a pocket pager and is programmed by a minimum of two Level II BASIC instructions or by four machine-language instructions. Within some constraints, the length of the beep is also software controllable. TBEEP is powered by a long-life battery (included) and is simply plugged in line with the AUX cable to the cassette.

Besides games, TBEEP can be used to signal when a long sort or data tape load is complete, or, by programming the ON ERROR

TRS-80 beeper.

Artec's ten-slot shielded motherboard.

The Beta-1.

The D101 and D201.

GOTO statement to go to a subroutine to activate the beeper, you can produce a beep even in the edit mode if, upon hitting RETURN, an error occurs. TBEEP measures 1¼ × 1¼ × 2¼ inches and is compatible with all TRS-80 Level II configurations, including disk. TBEEP sells for $18.95 (Calif. residents, add 6 percent).

Web Associates, PO Box 60-N, Monrovia CA 91016.

Silent Ten-Slot Motherboard

Artec Electronics, Inc., 605 Old County Rd., San Carlos CA 94070, has introduced a ten-slot version of its 16-slot, silent, totally shielded motherboard. Intended for use with the S-100 bus in compact systems with large memories, the ten-slot configuration allows the processor and peripherals to be condensed into a smaller package, without large amounts of spurious noise in the bus lines.

The board features the same one-eighth-inch FR4 glass epoxy construction and substantial ground traces as the 16-slot model. It also features Artec's PRC termination technique, which terminates each S-100 bus line in an optimum impedance without increasing the zero-state leading of the bus drivers. This technique eliminates the ringing commonly found on a bus, but does not limit the number of boards that can be placed on the bus.

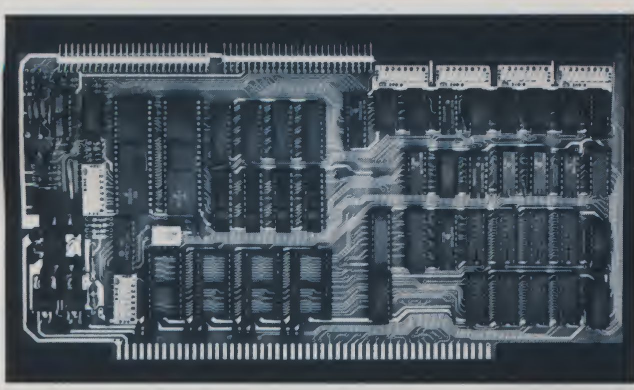
The 8×9.80 inches ten-slot board features Masterite connectors. It comes drilled to mount in the Imsai chassis and can be easily drilled to fit other chassis by the user. This shorter design means users do not waste money or space on unused motherboard capacity. The ten-slot motherboard is $115, assembled and tested.

The Switchboard

The latest design innovation by George Morrow is the Switchboard, an I/O board for S-100 systems. It has four parallel ports and two RS-232/TTY serial ports plus strobe and attention ports. In addition, there are options for 4K of RAM and 4K of EROM.

Every port is switch programmable for flexibility in interfacing various types of peripherals. Each parallel port can be switched for input or latched output. Both serial ports can be switched to any of 16 baud rates from 110 to 19K. Each strobe and attention port flip-flop can be switched for positive or negative pulsing. The eight I/O addresses of the Switchboard can be located on any boundary divisible by eight. Price is $199 kit and $259 assembled.

Thinker Toys, 1201 10th St., Berkeley CA 94710.



The Switchboard I/O board.

Tape Storage from Meca

A universal tape storage device that interfaces to most popular microcomputers, including non-S-100 bus systems, has been announced by Meca, PO Box 696, 7026 Old Woman's Spring Rd., Yucca Valley CA 92284. Called Beta-1, this unit plugs directly into a standard 8-bit parallel port. Serial port connection is offered as an option. The high-speed digital tape transport features random seek at more than 100 inches per second, with average access times in 10 seconds or less, and loading time at 8000 bits per second. An option is available to permit loading speed of 16,000 bits per second.

Employing the industry-standard phase-encoding technique, the Beta-1 is reported to be highly reliable. An internal 8035 microprocessor with a 1K byte program and high-level tape operating system assure easy-to-use operation. Storage capacity and performance compete favorably with more costly disk storage systems. Fully assembled and tested, the Beta-1 is priced at $399.

Darcom Telecommunication Products

Darcom, Inc., has recently announced its entry into the small computer periphery market with two telecommunication products: the D101 Data Coupler and the D201 Modem.

The D101 Data Coupler provides an automatic interface between a telephone line and customer-owned equipment, such as a modem or data terminal. It features standard Bell CBT functions, operates from a single +5 volt power supply and requires no adjustment. Ring indication, off-hook control and balanced data transmission are standard features. Price is $149.95.

The D201 Modem converts digital data to FSK suitable for telephone line transmission. It features automatic answer and disconnect, originate and answer modes and standard RS-232 interface. The modem operates full duplex to a 300 baud data rate and is compatible with the D101 Data Coupler. Its price is $249.95.

Darcom, Inc., 268 N. 115 Street, Omaha NE 68154.

TRS-80 Software

Get more out of your TRS-80 system with software from ACS Service, 2208 Dearborn Dr., Donelson TN 37214. The software is written for the TRS-80 only and includes company-backed, easy-to-understand documentation. Programs are stored on cassette tape or disk with complete documentation.

Some of the software currently available includes:

Z-80 Disassembler—Shows symbolic code for the machine instructions stored in memory in hexadecimal, ASCII representation and symbolic instructions with operands. Decodes all of the Z-80 instructions with Zilog mnemonics. Code can be reassembled using the TRS-80 Editor/Assembler. Will run on Level I with 16K RAM or Level II BASIC. Costs $20.

Data Base Manager—Lets you store any type of information with your own heading and file name and allows easy access later. Just give the computer some type of description to search the input string for words that will match what you have on file. If there is more than one file fitting your description, it will tell you so and ask which one you

(continued on page 21)

The Age of Affordable Personal Computing Has Finally Arrived.

Ohio Scientific has made a major breakthrough in small computer technology which dramatically reduces the cost of personal computers. By use of custom LSI micro circuits, we have managed to put a complete ultra high performance computer and all necessary interfaces, including the keyboard and power supply, on a single printed circuit board. This new computer actually has more features and higher performance than some home or personal computers that are selling today for up to $2000. It is more powerful than computer systems which cost over $20,000 in the early 1970's.

This new machine can entertain your whole family with spectacular video games and cartoons, made possible by its ultra high resolution graphics and super fast BASIC. It can help you with your personal finances and budget planning, made possible by its decimal arithmetic ability and cassette data storage capabilities. It can assist you in school or industry as an ultra powerful scientific calculator, made possible by its advanced scientific math functions and built-in "immediate" mode which allows complex problem solving without programming! This computer can actually entertain your children while it educates them in topics ranging from naming the Presidents of the United States to tutoring trigonometry all possible by its fast extended BASIC, graphics and data storage ability.

The machine can be economically expanded to assist in your business, remotely control your home, communicate with other computers and perform many other tasks via the broadest line of expansion accessories in the microcomputer industry.

This machine is super easy to use because it communicates naturally in BASIC, an English-like programming language. So you can easily instruct it or program it to do whatever you want, *but you don't have to.* You don't because it comes with a complete software library on cassette including programs for each application stated above. Ohio Scientific also offers you hundreds of inexpensive programs on ready-to-run cassettes. Program it yourself or just enjoy it; the choice is yours.

Ohio Scientific offers you this remarkable new computer two ways.

Challenger 1P $349
Fully packaged with power supply. Just plug in a video monitor or TV through an RF converter to be up and running.

Superboard II $279
For electronic buffs. Fully assembled and tested. Requires +5V. at 3 Amps and a video monitor or TV with RF converter to be up and running.

Standard Features

- Uses the ultra powerful 6502 microprocessor
- 8K Microsoft BASIC-in-ROM
  Full feature BASIC runs faster than currently available personal computers and all 8080-based business computers.
- 4K static RAM on board expandable to 8K
- Full 53-key keyboard with upper/lower case and user programmability
- Kansas City standard audio cassette interface for high reliability
- Full machine code monitor and I/O utilities in ROM
- Direct access video display has 1K of dedicated memory (besides 4K user memory), features upper case, lower case, graphics and gaming characters for an effective screen resolution of up to 256 by 256 points. Normal TV's with overscan display about 24 rows of 24 characters; without overscan up to 30 X 30 characters.

Extras

- Available expander board features 24K static RAM (additional), dual mini-floppy interface, port adapter for printer and modem and an OSI 48 line expansion interface.
- Assembler/editor and extended machine code monitor available.

ORDER FORM

Order direct or from your local Ohio Scientific dealer.

☐ I'm interested. Send me information on your:
☐ Personal Computers ☐ Business Systems ✓01
☐ Send me a Superboard II $279 enclosed
☐ Send me a Challenger 1P $349 enclosed
☐ Include 4 more K of RAM (8K Total) $69 more enclosed

Name ____________________

Address ____________________

City ____________ State ____________ Zip ____________

Payment by: BAC (VISA) ___ Master Charge ___ Money Order ___

Credit Card Account # ____________________

Expires ________ Interbank #(Master Charge) ____________

Ohio Residents add 4% Sales Tax

TOTAL CHARGED OR ENCLOSED ____________

All orders shipped insured UPS unless otherwise requested. FOB Aurora, OH

Interested in a bigger system? Ohio Scientific offers 15 other models of microcomputer systems ranging from single board units to 74 million byte hard disk systems.

OHIO SCIENTIFIC
America's Largest Full Line Microcomputer Company
1333 S. Chillicothe Road • Aurora, Ohio 44202 (216) 562-3101

✓ Reader Service—see page 147

BOOKS

Applications and Algorithms in Computer Science
C. William Gear
Science Research Assoc., Inc.
Chicago IL, 180 pages, $4.95

Sometimes, when you are sitting behind your computer terminal, do you feel like you are reinventing the wheel? "Surely somebody somewhere must have done this before!" I keep saying to myself.

How does a hobbyist who is turned on by the challenge of creating interesting and useful software obtain useful knowledge from the world of computer science? Do the software academics and professionals have things to teach micro-hobbyists with their little systems, or must hobbyists make their own way completely in the dark?

Hobbyist comrades, take heart. Here is a very readable book written by a competent academic (published by a subsidiary of IBM) that can teach us a lot about basic computer-science theory.

Applications and Algorithms in Computer Science is not a book for everybody. But if you are determined to make your computer useful, and earnestly want to learn how, this book will help you avoid a lot of learning the hard way. Best of all, it will give you confidence in the correctness of the problem-solving techniques you have discovered and chosen.

From the University of Illinois, Urbana, C. Wm. Gear has definitely written a textbook—or rather, part of one. The publisher calls it "one component of the modular text *Introduction to Computers, Structured Programming and Applications*." Preceding this volume is a "C" module, which introduces "Computers and Systems," and a "P" module, describing "Programming and Languages."

Three volumes discuss "Applications and Algorithms" in the "A" module to which this book belongs. Along with this one relating to "Computer Science," the other two cover "Science and Engineering" and "Business." A subsequent series of five volumes in the next module are called "Language Manuals," covering the use of FORTRAN, PASCAL, PL/1, BASIC and ALGOL.

The first two modules are considered prerequisites, but I found that with Dr. Lance Leventhal's excellent *Kilobaud* articles under my belt (especially "Why Structured Programming?" p. 84, Issue No. 14, February 1978), I was able to adequately understand the concepts presented. Algorithms are given in structured form for each new idea. Good text layout and illustrations clarify concepts, and step-by-step discussion in the body of the text makes sure that the reader gets the point. Nothing but praise is deserved for the author's thorough explanation of each idea presented.

The book is nicely divided into self-contained chapters, with all references to previously discussed subjects noted. As new material is introduced, a "Prerequisite Structure" diagram facing the first page shows what previous chapters should be mastered beforehand. Most moderately experienced hobbyists should have no difficulty with this book.

Beginning with a definition of five common types of computer algorithms (Direct Computation, Enumeration, Divide and Conquer, Iteration and Trial and Error), the text discusses the advantages and disadvantages of each with an example of an ideal application. Later chapters elaborate by starting with examples from real-world commercial and scientific applications, which helps put the whole discussion in context. The author's style is quite appealing, especially if you are one of those people who enjoy thinking about how complicated things work.

For example, the discussion about each of the three sorting techniques presented in chapter A3.2 (Selection Sort, Bubble Sort and Insertion Sort) includes the information needed to determine relatively how long each method will require depending upon the way the data is organized. This is the kind of discussion I have sought to help me gain confidence in the correctness of the methods I have chosen. And this is typical of the whole book, where the trade-offs between execution speed and memory requirements are candidly described.

Other subjects introduced, with typical algorithms given, are pointers, Chained lists, CAI (computer-aided instruction), Monte Carlo, Computer Simulation, Trees, Polish notation, Graphs and Critical-Path Problems, along with other math-related algorithms I found less interesting (Horner's Method, Linear Interpolation, Solution of Linear Equations and Numerical Error theory).

The quality of the algorithms given is shown in the fifth chapter, which discusses "KWIC Indexes." A KWIC (key word in context) Index of all the interesting *Kilobaud* articles you have accumulated would certainly facilitate searching through back issues. The index would alphabetize each "key" word in each title down the center of the page, with the rest of the words in the title all printed properly to the left and right of that key word. Each title would thus have as many entries in the index as there are significant words in it.

The task of preparing such a list would be enormous; only a computer could reasonably be expected to undertake it. A good explanation of the problem would be given with an overall algorithm followed by a very detailed one. From these, a resourceful programmer could put the pieces together given the peculiarities of his or her own system. Such a program for a micro should be highly marketable.

For my purposes as a computer hobbyist, however, this is where the weaknesses of this book begin to appear. Notwithstanding the string-handling power of 8K BASIC, implementing "structured" algorithms in an "unstructured" language is not easy.

I find I am still a long way from moving easily between theory and practice; I suspect that many hobbyists are in the same boat. Even when the theory behind a given routine is clearly understood, I still get hung up on the mechanics of making it do that on my machine! This book is full of magnificent new ideas for doing things I would like to do. If only, for our sake, the examples had been given in BASIC. (See following review.)

This is a book that can help the self-taught programmer grasp some of the basics of computer and information science, and make some real strides in technical capability. If you are good at moving from concept to application on your own, but sometimes feel your weakness in the basics, perhaps this is just the book you've been looking for.

Nelson R. Murphy
Rio Rancho NM

BASIC Language Manual
C. William Gear
Science Research Assoc., Inc.
Chicago IL, 67 pages

This book should be read alongside *Applications and Algorithms* (reviewed above) if your needs, like mine, are for examples given in BASIC. Dr. Gear has written this book, a companion manual in the series, to make it easier for us to understand how the principles of the structured programming techniques that he teaches in the other volume apply to a BASIC-language environment.

The first ten pages give a cursory overview of BASIC statements for persons familiar with programming in other languages. The material in the next section is good for learning how to implement in BASIC the important constructs of the structured programming technique. For example, on page 14, the author gives a general template for the WHILE . . . DO indefinite iteration loop that is not built directly into the syntax of BASIC. Similarly, the IF . . . THEN . . . ELSE construct is also demonstrated on page 17, for those of us not lucky enough to have that feature in our version of BASIC.

The material in the second section is readable on its own, but it was intended for college students working through a course that might be called "Introduction to Computers, Structured Programming and Applications." That is the title of the series that both of these books come from. An earlier book in the series, *Programming and Languages*, has a much more complete discussion of the concepts presented, with the examples and illustrations all given in a pseudo-language the author has developed, and which looks a lot like PASCAL or ALGOL. This manual then transliterates that universal non-language into the syntax of BASIC for several of the examples, so that the basic principles can be demonstrated on computers that use BASIC.

The same is also true for the

final sections of the manual, only this time the correspondence is with the *Applications and Algorithms* volume already reviewed. And that is where the real value of this workbook lies.

Starting with the full description in *Applications and Algorithms* of implementing a binary search, for example, the reader can then turn to the *BASIC Language Manual* and find a directly parallel BASIC version of that algorithm demonstrating how to apply that technique on his or her machine.

This manual's BASIC version of the Insertion Sort algorithm given in the first book is, alone, worth the manual's price, but for the kind of operations I have been trying to perform on my microcomputer, the gem is the BASIC program given for implementing linked or chained lists. (I didn't know what a chained list was before I read the first volume; after finding out, I became frustrated trying to figure out a way to implement them in BASIC. Hence, it was to my utter delight to see the concept so beautifully illustrated in my own "mother tongue," BASIC.)

Obviously, the purpose of the manual is not to illustrate everything, but rather to teach the student how to move from the concepts presented in the first volume to the working implementation of it in a computer running BASIC. While one might wish that all the algorithms presented in the first volume were transliterated in this companion manual, that would not serve the learning process as well as the to do-it-yourself method. Without question, there is enough material in this manual to demonstrate clearly how to go about making any of the other algorithms work in BASIC, too.

That makes this book an invaluable addition to one's knowledge about good programming techniques.

Nelson R. Murphy
Rio Rancho NM

Low-Cost, Personal Computer-Based Investment Decision Systems
Jerry Felsen, PhD
CDS Publishing Co.
New York, 1978, 169 pp.

How would you like a microcomputer for a business partner? If so, read on. Jerry Felsen's *Low-Cost, Personal Computer-Based Investment Decisions Systems* is written as a guide for the person unfamiliar with computers who is interested in using a computer as a partner in trying to buck the odds playing the stock market. Felsen explains the role the computer can play in aiding the speculator to make profitable investments. Felsen advocates the investor's using his intellect and intuition with the extremely fast computational powers and perfect memory of the computer. In this respect the computer does not replace the investor, but augments his intellect with its own capabilities.

The book is divided into two sections. The first is devoted to explaining the concepts of cybernetics, artificial intelligence and pattern recognition. Also covered are some of the basic elements of current market-investment theory. Only the general role of the computer and its part in the "man-computer synergism" are covered. No specifics of actual software or computer algorithms are given. Readers interested in the actual details of software for computer-based investment systems are referred to two of Felsen's earlier books: *Cybernetic Approach to Stock Market Analysis Versus Efficient Market Theory* and *Decision Making under Uncertainty: an Artificial Intelligence Approach.*

Section two, which comprises the majority of the book, covers the hardware end of current microcomputer systems. Everything is covered, from CPUs, various solid-state memories, disk and tape storage and the necessary input and output devices needed for a complete system. Everything is explained in a manner suitable for a person who does not have a background in computer science.

A detailed plan for designing, acquiring and implementing a functional, cost-effective computer system is presented. Felsen discusses the pros and cons of buying or leasing a computer system, developing or purchasing software from professional software houses, maintenance contracts, etc. He covers the entire sphere of decisions one must make when implementing any computer system. The practical requirements for any computer system (in relation to the problem to be solved) are essentially the same. Therefore, this portion of the book is particularly valuable and recommended to anyone planning to purchase a computer system regardless of the end use in mind.

In summary, this book is an excellent source of information for the person who intends to buy a computer system. It will provide the experienced computerist with the philosophies pertaining to designing a computer-based investment system. Those particularly interested in the investment-decision side of the book will no doubt also be interested in the two books previously mentioned in this review.

Richard Dykema
***Kilobaud* Staff**

COMPUTER CLINIC

This column is designed for readers to present questions and problems to be answered and solved by readers. If you wish to respond to any of the questions here, please contact the readers with problems personally at their respective address(es). Problems and questions sent to *Kilobaud MICROCOMPUTING* should be typed, double-spaced. —Editors.

I acquired an Imsai PCS 80/10 locally because I thought buying locally was safe; however, as it turns out, my vendor can't, or won't, supply me with information to help me out with the predicament he got me into.

My machine is the Imsai 8080 with front-panel switches, etc. My problem is that the seller sold me a Processor Technology 3P+S board instead of an Imsai S10 interface, and it appears as though he has stuck me with it.

I have a Data Set 201 for my RS 232 I/O device, and I have a Teletype Model 33 ASR for serial I/O. I know all the devices are operating.

The Imsai PCS 80/10 user manual has a good software program listed, but I am unable to use it because I can't load it from the paper tape Imsai supplies.

The problem, I believe, is in the bootstrap loader.

Unless I can get the paper-tape software into memory from my 33 ASR, I shall remain on zero ground.

Mike S. Mudray
47 Elviss Cr.
Brandon Manitoba
Canada R7B 2K8

I just purchased a TRS-80 microcomputer with Level I and 4K memory and ran into what seemed an interesting problem. My Level I BASIC will not handle a dimension statement: for example, DIM D(121) in the October 1978 issue, page 40.

I would like to know if there is any way to write around this type of program statement, provided you have enough program space to do so. Also, is there any way I can work around a PEEK command such as A = PEEK (5888) AND 1 as in the November 1978 issue, page 34?

Edward P. Jelf, Jr.
224 Fairdale Drive
Lexington KY 40505

I have two Sperry Rand Univac "Uniscope 100" CRT terminals. However, I have no technical info on them. Do you have information on the identification of pins in the output cable as to function and voltage levels?

It's my understanding that there are two types of interface for this unit. The RS 232-C and a "multiplexed" type. I have probably got the multiplexed type as these units were *not* connected to a modem as would be common for the RS 232-C model.

I hope to be able to obtain sufficient information to enable me to build an interface compatible with some other equipment I have.

Milton E. Miermaster
6030 W. 87th St.
Overland Park KS 66207

I have a very fine SWTP 6800 system with 20K memory, CRT and two cassettes, neither of which will accept any of the currently standard recording formats. I recently tried to order SWTP's BASIC in punched paper tape, which I can read, but it is no longer available from them. One possible solution would be to get along with Tom Pittman's Tiny BASIC (a good starting point), but after making some revisions and programming around the

minor problems of this version of BASIC, I would like to see how someone else has done it and expand my horizons as a programmer. Please contact me if you have information about the price of a copy of the tape and user's manual or rental of the same.

Gordon J. Tennyck
667 Shunpike Rd.
Green Village NJ 07935

I need some information on what company has hardware and software to interface the Apple II computer to send and receive CW and RTTY. I have been trying to find information all over Houston.

Doyle Hansel
3810 Law 2
Houston TX 77005

For about four years now I have owned a Mits 8800. For a long time, I was the only kid on the block with a computer. I was happy with my own operating system and machine-language programs. But now I would like to have some compatibility with the new kids on the block. I would like to know if anyone puts out a small BASIC on ROM. I would rather not have Tiny BASIC; a small Level II would be very nice. I have 9K of free memory to work with.

Dennis Rinaldo
28024 Tracy Rd.
Walbridge OH 43465

I ordered an entire system from Technico in January 1978 (TMS 9900 CPU, 32K memory, chassis, power supply, super BASIC). The system is nonoperational, and I cannot get adequate response from Technico. The product itself appears to be of high quality, but the service is wretched.

I need help badly, and feel that I am not the only one. If you have information about *any* hardware or software fixes, or where to buy parts, please contact me.

For those who need help, send some large, self-addressed stamped envelopes, and I will give you all the data I get and will pass on and/or answer any questions you may have.

M. B. Kelly
10 Bougainvillea Sumay
U. S. NAVSTA
FPO San Francisco 96630

Can anything be done for us people who have non-North Star mini-drive systems? *I* have re-

(continued on page 21)

LETTERS

Nix to 'Nics

I enjoyed Ralph Wells' final Troubleshooters' Corner (November 1978). I was also pleased to catch his supporting comment about the service technicians of Tektronix. It may not have been intended as such, and I'll accept the plug for all of Tektronix's field-service technicians. I would like to mention that the correct spelling is with IX at the end and not ICS as most people have the tendency to write. We are not put out by this, however, as we know that our pride and quality precede us in the enthusiasm our customers have and share with others.

John R. Carter, Sr.
Systems Analyst
Tektronix, Inc.
Beaverton OR

Home Brew Is Where the Heart Is

W. B. Chess of Monsey NY wrote that in "Home-Brew Z-80 System" by Norm E. Thagard (July 1978, p. 81), "In Fig. 1, IC3 has pins 4 and 1 returned to ground. They should go to Vcc. A 7474 will toggle if set and clear are logic 1. If set and clear are logic 0, the chip will not function." Norm's reply follows.—Editors.

Mr. Chess is correct about the flip-flop. The pins in question are tied to Vcc on the prototype. In checking my original schematic, the pins are erroneously shown grounded there, too. Who knows what gremlin caused me to draw it that way?

A reader has pointed out to me a sin of omission on the front panel schematic. The 74193 counters, ICs 11-14, need to have pin 14 (clear pin) grounded. They are grounded on the prototype, but the schematic failed to show it. This is necessary because of the active high clear feature on the 74193. Thus, the counters will always be cleared to 0 if pin 14 is allowed to float high.

Finally, let me mention a potential problem that was not manifest with the original 4K memory boards described in the July article but which appeared when I built a 16K circuit. The problem is caused by the $\overline{\text{WRITE}}$ pulse arriving on out (port) operations before the SOUT signal can prevent the memory from responding. Thus an OUT A, 00 can cause a write to memory at the address corresponding to the accumulator contents as high byte and port address as low byte. If this is a problem, a simple cure is to delay $\overline{\text{PWR}}$ at the CPU board by inserting two inverter-gates or buffer gates between pin 22 of the Z-80 chip and pin 12 of IC7. Note that pin 22 should still go directly to pin 9 of IC6.

Anyone needing help or information on the circuits in the article is welcome to write me at Mail Code CB, Johnson Space Center, Houston TX 77058, or, better, 528 E. Castleharbour, Friendswood TX 77546.

Norm Thagard
Mission Specialist
Astronaut Candidate

Looking Before Leaping

Recently we received a couple of letters alleging that the program in "Computer-Generated Signs" by Joe Roehrig (August 1978, p. 90) was "full of mistakes." Joe Roehrig's reply, which follows, provides some good food for thought.—Editors.

Regarding the two letters, nothing is wrong with the program. As you know, and the two readers apparently don't, there are many versions of BASIC. My program used North Star BASIC, in which L$(1,3) could equal ABC; L$(2,3) could equal BC.

These two readers have a Radio Shack and possibly an Altair version. Here, subscripts are used as if the variables were numeric.

Joe Roehrig
Middle Village NY

Conversion

Although it is not my MO to write to magazines, I make exception here because of the unusual quality you produce. In "Universal Number Converter" (November 1978, p. 67), the program has one small bug: the accumulator NO$ will not clear and it will produce undesired numbers after repeat runs. It can be rectified very easily by inserting a CLR immediately after line 410.

I like the format of your magazine. Keep up the good work.

W. T. Mallison, Jr.
Rocky Mount NC

Regarding the letter from W. T. Mallison concerning problems with my "Universal Number Converter": The fix that Mr. Mallison suggests seems to me to have no effect on the running of my program. However, in some instances errors that are caused by round-off errors in the BASIC floating-point routine seem to occur. The modifications to correct the accuracy problem being encountered are:

```
10900 DEC = DEC + INT(K%*BI%↑PWR
      % + .5)
11310 XX = INT(BO%↑J + .5)
11600 DEC = INT(DEC – CH%*XX + .5)
```

I hope these patches will fix the problem. Sorry for any inconvenience I have caused you and your subscribers.

Easton Beymer
Huntsville TX

PETting the Record Straight

I guess it's a reflection of the effectiveness of *Kilobaud MICROCOMPUTING* that I'm being swamped by mail from the item in Publisher's Remarks, page 7, of the November 1978 issue. By giving my address, you effectively credit me with being the publisher of "PET User Notes," which is much more than I deserve.

The *real* publisher of "PET User Notes" is Gene Beals, who is doing a great job, essential because Commodore has been very slow in supporting their otherwise fine product with adequate documentation. Gene's address is Pet User Group, Box 371, Montgomeryville PA 18936. A subscription is $6 for six issues, and easily worth twice the price.

My own PETitle is software librarian for the PET user's subgroup of the Amateur Computer Group of New Jersey. We have a rather large club library of cassette tapes available for copying on our meeting nights (fourth Fridays) and a collection of PET technical bulletins and other useful papers available either at the meetings or by mail.

The people who write as a result of your item will get an assortment of our paper, and be referred to Gene. However, I hope you'll be able to either print this letter, or at least a correction, as the postage alone could break us.

Roy O'Brien
South Bound Brook NJ

For more information on the PET, see this and last month's PET-pourri.—Editors.

Kilophile

I hope you will always keep *Kilobaud MICROCOMPUTING* a multilevel publication as you indicated in your reply to Gary Koehler (December 1978, pp. 16-17). What many "advanced" computerists don't realize is that ever-increasing numbers of beginners like me are just beginning the trials and tribulations they went through long ago. Each issue is the very *first* issue for many, many newcomers, and you are doing us all a great service by providing a balanced mix.

To carry the point a bit further, I urge people such as Mr. Koehler to make "a useful and unique contribution" (using his words) by putting advanced skills and techniques (which he seems to feel are possessed by everyone) into short, understandable modules of information and write articles for *Kilobaud MICROCOMPUTING.*

I am a stamp dealer and work with collectors at all levels of understanding. Sometimes I find it is through helping the novice that I learn the most about the limits of my own knowledge. Now that I have my own computer, I have been abandoned by the programmer I counted on to help me, so I, for one, am grateful that *Kilobaud MICROCOMPUTING* is still there with a balance that will both help me now—and later.

Jeff Purser
Danbury CT

Good Ol' BASIC

I have just finished rereading the Publisher's Remarks column in the December 1978 issue of *Kilobaud.* (If there is anyone who can inspire someone to think to the future, Wayne Green can.) It got me so excited, my mind went into high gear and nearly got stripped. All the thinking about education and how to communicate via the video medium that I have been doing the last ten years or so Wayne put into a nutshell.

I am very interested in Instant Software packages, but I have a

problem there, as probably others have. My system is a KIM-1, which now has a graphics board and nothing else. I program in machine language. You listed a lot of systems that use various high-level languages. Question number one is, do I, or can I, fit into your scheme as I am? Number two: Some day, as the budget allows, I'll be implementing a higher-level language. What language should I get into, and what version should I use that will best fit into your Instant Software program? BASIC does seem to be a popular language, but is it the best? It is a most confusing question. I want to go toward PASCAL, but it seems to need a lot of memory and is not now a hobby language. BASIC seems to be the most widespread hobby language, and a lot of small computers use it, but then, what version, or does it matter? Other easy languages seem to be coming along, but they are not widespread enough. What a mess. We do indeed need a strong voice to lead the way for some kind of standardization. Can you shed some light on this for me?

Anthony T. Scarpelli
N. Windham ME

Yes, I can answer your question, but you'll probably want to argue with me. The answer is quite simple. It doesn't make much difference how good PASCAL or FORTRAN are—BASIC is what virtually all *microcomputers are using, so it is really counterproductive to go any other route.*

When one gets down to the nitty-gritty, the language isn't as important as many other factors. Sure, PASCAL can do some things better . . . so can FORTRAN . . . and all other languages. Until the day arrives when we can do math with FORTRAN . . . business with COBOL . . . and use each language for what it does best, with compilers on tap for all popular languages on all microcomputers, we're stuck with BASIC.

For now, you'd better accept this fact of microcomputing life and stick with BASIC.—Wayne.

Hitting the Spot

Your December issue really hit the spot. Lewis Tarnopol's "The Care and Feeding of Cassette Tapes" really hit a vital need. I wish that he had been able to mention in a positive sense more than two satisfactory brands, but I could see legal problems if he had done so.

The most useful article was Barry A. Lewis's "Deep, Dark Secrets of the TRS-80 (Level I)." I would like to see Radio Shack (or someone) write an assembly-language programming textbook for the TRS-80 as good as and in the same format as the Level I instruction manual. Machine-language programming will be even more necessary if hobbyists attempt to interface exotic peripherals to their TRS-80s.

I'd like to see similar articles.

James H. Sheats
East Point GA

Flips Over Reverse-Side Article

The article, "Scratched Diskette," on page 106 of the December issue is great. I've been curious about using the reverse side of the diskette. I tried it with a full-size disk and it works perfectly. I was also happy to see the letter praising George Morrow. He is, without a doubt, one of the outstanding gentlemen of personal computing. (I like the term Personal Computing!)

Rod Hallen
Tombstone AZ

A Rotten Apple?

If damned with faint praise is the name of the game, "The Remarkable Apple II" (November 1978, p. 62) is surely a "winner." Hopefully, competition in the marketplace will eliminate products like this that are so bereft of features offered by lower-priced and higher-performance machines such as the Level II TRS-80. Caveat Emptor!

Robert M. Richardson
Chautaqua, NY

Apple owners—what's your response?—Editors.

OSI Owners, Unite!

As a new subscriber I just received the December issue of *Kilobaud*. Looking through the index for 1978, I see no articles about the OSI computer. If you want me to renew my subscription, sometime within the next 11 issues I want to see at least one article about the computer I am using.

E. Morris
Midland MI

How about it, OSI owners? Are you going to let all those other micros hog the glory? If you have an interesting mod or other useful information, write it up for Kilobaud MICROCOMPUTING. *It's easy, and it might just pay for that new peripheral.—Editors.*

From Guam

I've recently decided to buy a computer and have been looking through your back issues to become familiar with systems and jargon. In the April 1978 issue I ran across a letter from Ted King, which reminded me of my experience in trying to see a computer in action at a store.

Here on the island of Guam, only one place sells computers . . . a hobby shop that deals with Radio Shack. After seeing a stateside-produced TV ad for Radio Shack's computer (paid for by the local dealer), I rode my motorcycle down to the store to look at their demo and to ask some questions. After finally getting the attention of the salesman (he was busy selling a 59¢ part to another customer), I turned on the computer. However, the salesman immediately turned it off again, explaining that the computer and a radio blaring from across the room interfered with each other. When I suggested turning off the radio instead, he simply stared at me. I was so mad, I stalked out of the store, jumped onto my bike and roared off. Unfortunately, in my anger I neglected to unlock the steering lock on the front tire so I rode around in a very tight circle until I fell over.

I'm now planning a trip to Los Angeles in January so I can look at some computer stores there; hope I can find a store that doesn't have radios that interfere with computers.

Keep up the good work.

Samuel E. Rhoads
Agana Guam

From Germany

With much interest, I read in the September 1978 issue the Troubleshooters' Corner concerning the problem of adding the PIA 6520 or the VIA 6522 to the Apple bus.

In our institute, we have been using a KIM-1 (+4K bytes) for a long time, and we got an Apple II (with 16K) in March 1978. The first thing I did was develop a little double-sided PC board, which plugs into the Apple bus and has a PIA 6520 on it to put the Apple and the KIM together.

It worked from the beginning. With a little service routine, I can send programs from the Apple to KIM and vice versa. The reason is, I have a very fine machine-code editor, running on the KIM, which allows me to insert, delete or change program lines with an automatic correction of the relative branch instructions. Therefore, I am writing my programs with the miniassembler of the Apple; for correcting I use my KIM.

Maybe you are interested in what we are doing with our micros. The KIM is used to control stepping motors, time measurement and data handling. The Apple serves as a desk calculator for the evaluation of numerical results from experiments. I am testing a Fourier analysis program and a program for the integration of differential equations, both written in BASIC. Finally, I just began to write a program for developing the layout of a PC board, using the HIRES graphic of the Apple to get all the information punched on tape for getting a plot on our large Univac.

Ekkehard Flogel
Universitat Karlsruhe
Institut fur Mechanik
Karlsruhe Germany

S-W-T-P

The October 1978 issue of *Kilobaud* contains a letter from Dexter French, on page 21, regarding a way to trick SWTP 8K BASIC into using a smaller block of memory than actually available on a system. Since I am using SWTP V2.3 8K BASIC I used the hint from Mr. French as a starting point to find the place in V2.3 BASIC that would accomplish the same trick. The information supplied in his letter applies exactly, except the magic location to be changed is $0CAC.

I share the appreciation of Mr. French for the coverage of SWTP systems. I would like to use this opportunity to appeal to other 6800 users to consider sharing their 6800 secrets. As an example—has anyone found a fix for the annoying problem of not being able to begin a variable name with the letter E if it is immediately preceded by a line number?

I have two rather simple-minded solutions: Put any other statement in front of it and make it a multiple line statement or, more simply, don't use variable names beginning with E.

Joseph J. O'Loughlin III
Huron OH

PET-POURRI

(from page 10)

first *two* letters are the *only* significant letters to the PET.) Later in the program I use variables SCREENNOW and SCREENOLD. The PET cannot distinguish between them but they help us keep track of our program.

Lines 100 to 170 draw a border line around your screen. This is nice for the user to see where the sides are and will be used again in line 610 to make sure the ball doesn't run off the screen. Note: As you go across the top and bottom, your position increases by 1; but as you go down, your position increases by 40. Thus the side walls are multiplied by 40 as I go down.

Lines 300 to 320 get the direction to move. Note: We GET DI$. If you input GET DI, the PET expects a number. If you then hit any other key (such as a T), you get the cryptic message REDO FROM START. Once we have the string variable we convert it to a numeric variable in line 310. The VAL command returns the VALUE of the string. The value of letters and nonnumeric characters is always 0. Thus line 320 tells the PET to GET another character if the value is 0. Yes, the number 0 has the value of 0 too, but we are looking only for the numbers 1-9.

Line 400 is the ON . . . GOSUB command. It works like this:

```
ON A GOSUB 1, 2, 3
```

The value of A determines where the PET will go next. If A is 1, it goes to the line number immediately following the word GOSUB. If A is 2, it goes to the second line number listed after the word GOSUB. If the value of A is not an integer, the fractional part (right of the decimal point) is eliminated. Thus 2.458 is seen as 2. To make line 400 easier to follow, I use GOSUB 1000 if DI is 1, 2000 if DI is 2 and so on through 9000 if DI is 9.

ADJUST is the variable used to update our current position on the screen. Line 40 puts us in the center to start. If we hit a 2, the ON . . . GOSUB sends us to line 2000. Since 2 means DOWN, we must add 40 to SCREEN to get the position 1 unit down. So we say ADJUST = 40. Note: Since this is a subroutine we must end it with a RETURN. If we want to go left, we hit the 4, which sends us to line 4000. To go 1 unit left we must add a −1 to our current position, so ADJUST = −1. Thus line 400 with the subroutines in lines 1000 to 9000 tells us how to adjust our position.

Lines 500 to 720 move the ball and leave a trail behind so we can tell where we have been. First, line 500 POKEs a trail into where we are now. Line 710 POKEs a ball into where we are going. Lines 600 and 610 check to make sure we didn't hit a wall. If we did, we skip line 700, which updates our position, and just go to line 710 to POKE a ball back in the old position, which we did not update. To add sound, we simply add the three lines in Example 1. Many other things can be added or modified.

Next month the example program will show how to have your PET actually change your program for you *while it is running*, so you have something to look forward to.

Problems

If you have problems loading a tape without a LOAD ERROR, and your tape heads have been cleaned and demagnetized, try this: Put the cassette inside a plastic bag and store it in your refrigerator (*not* the freezer) for one hour. Take it out and try to load it in your PET. It may take a couple of tries. Once it is loaded save it on a new tape. Throw the faulty tape away or you may be tempted to use it again.

If your keytops seem to be wearing fast, coat the tops of each key with clear nail polish before they wear too badly.

If you have any problems or questions, hardware or software, please write to me at 1929 Northport Dr., Room 6, Madison WI 53704. I also look forward to your ideas and suggestions for this column.

```
302  POKE59467,16 :POKE59466,9 :POKE59464,222
304  FOR NOTE=1 TO 15: NEXT NOTE
306  POKE59467,0 :POKE59466,0 :POKE59464,0
```

Example 1.

COMPUTER CLINIC

(from page 18)

cently obtained a control board and SA400 mini-drive from S. D. Computer Products. They supplied CP/M—and as an extra, E-BASIC for no charge. I have a TDL monitor board so I had to drop the disk ROM down to E000H.

However, I now have a software-exchange problem. To get software from full-size disks, I have to load it into memory and copy it out on tape from a computer store, then load it into my system by a reversal of the process. Could somebody knowledgeable about the 1771 chip write software to enable one to read CP/M from a hard-sectored disk system? Meanwhile, I would like to set up an exchange group for software for the S. D. Sales superfloppy. Please send your name, address and software available, if any, so that I can compose a list to be returned to each user.

John J. Momohan
215 Passaic Ave. #4-J
Passaic NJ 07055

PUBLISHER'S REMARKS

(from page 8)

write 'em and Instant Software will sell 'em. Please make sure that your programs are ready for publishing, not just in rough draft. They should be checked and rechecked. It is too late to debug things after the first ten thousand have been shipped.

Now It's Color?

Anyone who has seen a Compucolor exhibited at a show knows what a grabber a full-color computer display can be. This is true of the Apple, too, though its use of color appears to be a bit more limited than Compucolor's.

In our development work on Apple programs we see the spectacular improvement in some programs when they are rendered in color. The Golf program is fun in black and white, but it is much more exciting in color.

When you consider the applications for color in a business environment, it seems unavoidable that all systems will have to go full color in the near future. Two years from now we may consider black and white displays as old fashioned as black and white television.

As we get into this new dimension we'll find more uses for it. I can certainly imagine an accounts-receivable display where 90-day accounts are in red, 60-day in orange, 30-day in yellow, less than 30-day in green, etc. It would give a fast visual appreciation of the aging of accounts.

The same idea could be applied to an inventory display, with back-ordered items in red, overage items in blue, etc. The normal black and white terminal has just two colors you can use: gray and white, with gray usually being used to indicate a protected field. With color you could easily show protected fields, important information, unimportant and more.

It is my understanding that we will be seeing more and more color in our systems . . . our larger-selling systems.

NEW PRODUCTS

(from page 14)

want. It requires Level II and mini disk and 16K RAM. Costs $20.

Inventory—A machine-language program to enter your inventory listing complete with all related information for fast retrieval. You can create, save, retrieve, modify and sort any type of information. It requires Level I 4K RAM or Level II 4K RAM. Price is $20.

File Handler System—Sets up a high-level filing system for input or retrieval and adding any kind of data that fits the program you are writing. You must add the input and output lines to fit your program to give you large-computer, file-handling capabilities. Requires Level II 16K RAM and mini disk. Program price is $29.95.

EPROM Programmer

The EP-2A-79 EPROM Programmer spans eight microcom-

The EP-2A-79.

Bridge Challenger on the PET.

puters and programs 1K, 2K and 4K EPROMs. Software for programming and verifying programming is available for the 6800, 8080, Z-80, 8085, 6502 (KIM-1), F-8, 1802 and 2650 based microcomputers.

Packaged in a sloping panel aluminum case, the unit connects to the microcomputer with a 14-pin ribbon cable through 1 1/2 I/O ports. Software requires approximately 256 bytes of RAM and includes instructions on how to relocate. Personality modules that plug into the front panel-mounted socket are available for programming the 2708, 2716, TMS 2716, 2732, TMS 2708 and TMS 2532 EPROMs. Power requirements are 115 V ac 50/60 Hz at 15 Watts.

The EP-2A-79 is priced at $145, which includes one set of software. Personality modules are priced at $15, except the personality modules for the 2732 and TMS 2532, which are $25.

Optimal Technology, Inc., Blue Wood 127, Earlysville VA 22936.

X-Y Plotter Unit

Sylvanhills Laboratory, Inc., has introduced a new X-Y plotter unit. This unit includes a plotter, drawing surface, electronics and power supply completely assembled and ready for interface to any eight-bit TTL parallel port. The pen holder accepts any writing instrument or stylus 7-11 mm diameter; it is encoded for 0.01-inch/pulse, but 0.005-inch is optional. Pen travel speed is 2.5 inches/sec maximum with 24 volt supply. A BASIC 8080 software program is included in the owner's manual.

Applications include architectural, mechanical and schematic drawings; PC board artwork; positioning of small objects; computer-generated art; games; and many others. Two models of the unit, with plotter, console and power supply, are available: Unit-1—11 × 17 inches drawing area, $1049, and Unit-2— 17 × 22 inches drawing area, $1249. Plotters are also available in kit form with console and power supply priced separately.

The plot driver software is now available as ASCII source files on paper tape, CP/M small disk formats. TEI and Cromemco small disk formats are also available. Both the BASIC and assembler source are provided and come with more comments to guide you in making source modifications.

Sylvanhills Laboratory, Inc., PO Box 646, Pittsburg KS.

Bridge for PET, TRS-80, Apple

No longer do you need four people to play bridge. With the Bridge Challenger program from Personal Software, PO Box 136, Cambridge MA 02138, you and the dummy can play four-person Contract Bridge against the computer. The program, designed for 8K PETs, 16K Level II TRS-80s and 16K Apples, will deal hands at random or according to your criterion for high card points, and you can save hands on cassette and reload them for later play. You can review tricks, rotate hands East-West, shuffle only the defense hands or replay hands when the cards are known. Bridge Challenger is priced at $14.95.

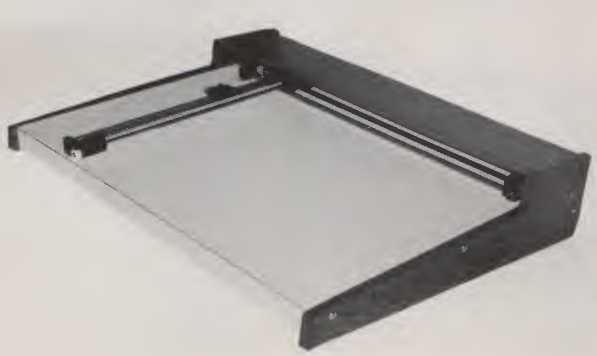
Line Printer from Heath

The WH-14 Line Printer is designed for use with the H8 and H11A computer systems (and others) using a standard serial interface. It prints standard 96-character ASCII set (upper and lowercase) on a 5×7 dot matrix print head with a maximum instantaneous print speed of 135 characters per second. Line spacing is six lines per inch (eight lines per inch software-selectable) with selectable line length of 80, 96 or 132 characters. Baud rate is also selectable from 110-9600.

The WH-14 uses 0.5 inch wide nylon inked ribbon on 2 inch spools. Adjustable width sprocket feed allows the use of edge-punched fan-fold paper forms from 2.5 inches to 9.5 inches wide having a maximum thickness of 0.006 inch.

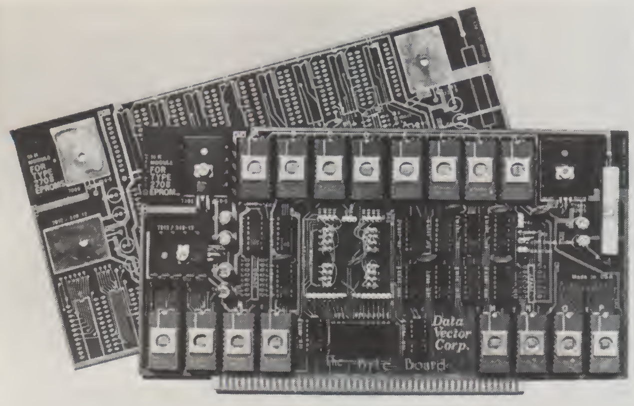
The WH-14 connects to the H8 or H11A computer via a standard RS-232C serial interface of 20 mA current loop. Handshaking is provided by reverse data channel or busy control signal. A 25-pin male EIA connector is provided for hookup, and a paper rack is included at no extra cost. A kit version (the H14) of the WH-14 will be available in the near future. Assembled and tested, the WH-14 costs $895.

Heath Company, Department 350-820, Benton Harbor MI 49022.



Sylvanhills' Plotter.

The H14.



The Byte Board.

Protect Your TRS-80

Now you can dust-proof your TRS-80 system with colorful fabric dust covers from August Automation, 42 West St., Westboro MA 01581.

Static is not a problem for these covers. All are machine-washable and pre-shrunk. Color options are red, maroon, green or gold velour; blue or brown corduroy; and blue denim. $19.95 for each 3-piece set.

EPROM Module from Data Vector

The Byte Board is a new EPROM module for the S-100 bus from Data Vector Corporation, PO Box 3141, Burbank CA 91504. The unit accepts up to sixteen 2708 EPROMs, incorporates a Power-On Jump capability, provides for wait state generation for slow memories and possesses extreme addressing flexibility.

Each EPROM is individually addressable on any 1K boundary and may be placed anywhere in the computer's address space, irrespective of where any of the other EPROMs are placed; unused EPROM locations do not take up memory address space. Additionally, all user selectable options are clearly silk-screened onto the board, allowing the state of all options, including the address assigned to each EPROM, to be read directly from the board without referring to the manual.

The module is solder masked to minimize solder bridges during assembly, is fully socketed, has two spare IC pads for circuitry and comes with a 20-page owner's manual.

The Byte Board, without EPROMs, may be purchased assembled and tested for $99 or in kit form for $69. Fully loaded with 16 EPROMs, the module is $243 assembled or $213 in kit form. EPROMs may be purchased separately.

Business/Accounting Packages

Two new business and accounting program packages for Wang computer users are now available from Data Train, Inc., 840 NW 6th St., Grants Pass OR 97526.

DTI Payroll #401 and #402 are designed to run on Wang dual mini, dual floppy and hard disk systems containing a minimum of one flexible disk drive with 8 to 16K memory.

The Payroll product has been field tested by practicing accountants and businesses and designed with the beginning computer user in mind; an accountant background is not necessary for operation, and training time is minimal with the provided reference manuals. Features include user selection of storage devices, easy user support of all tax tables, concise accounting, departmental and labor reports and special reports under user design and format control. The #401 costs $750; the #402 is $900.

The General Ledger 101 operates on Wang T and PCS II models with various disk units. The package features all necessary accounting reports and audit trail and user selection to format the financial statements to meet internal or customer needs. Format control is provided for Balance Sheets, P & L and Change in Financial Position, which may contain up to eight columns of data.

Special reports may be designed and obtained through the systems report writer feature. G/L 101 will process multi-divisional departmental companies and provide appropriate reports, as well as summarize journal entries for posting and/or reporting. The software product requires 8K to 16K memory and dual mini or floppy diskettes or hard disk with a single floppy available. G/L 101 costs $850.

Cassette Control Unit

As TRS-80 owners know only too well, the small plugs and jacks that are standard equipment on cassette tape recorders were never designed for the constant use they must endure as part of a microcomputer system. In addition, the continual plugging and unplugging required to use the rewind and fast-forward functions is tiresome and not very elegant. The Micro-Mega Cassette Control Unit puts the solutions to these and other problems into one tiny box.

With the Micro-Mega inserted between your TRS-80 and its cassette recorder, you'll never again have to unplug the "remote" cable to rewind a program tape. A single switch on the unit gives you complete manual control over the recorder.

The Micro-Mega also features a built-in speaker (with volume control) that allows you to monitor the data on your tapes, greatly simplifying the task of finding the beginning or end of a program. This function of the Micro-Mega is especially useful if you store more than one program on each side of a cassette and your recorder lacks a tape counter. The monitor may be switched off when it's not needed.

As a final plus, the Micro-Mega reduces the TRS-80's 60 Hz hum problem by the simple expedient of terminating the "ear" cable ground connection inside the unit's case. This practice eliminates one of the ground loops that contribute to the hum problem. The Micro-Mega is ruggedly constructed and can be plugged into your TRS-80 system without modification to either the computer or the recorder.

Micro-Mega, PO Box 6265, Arlington VA 22206.

8080 Checkers

Sharpen up your checkers-playing skill with a challenging new program from TCD Incorporated, PO Box 58742, Houston TX 77058. 8080 Checkers can be set to play at two different levels of difficulty (four or six moves ahead) for both the beginner and the advanced player. At level four the program will respond in less than four seconds, and at level six the program will typically respond in less than 60 seconds and rarely more than 120 seconds.

The program will run on an 8080/Z-80 computer with 12K RAM and a memory-mapping display such as the SOL, VDM-1 or TRS-80. The software is distributed on CUTS cassette tape (orged at 0) and on North Star diskette (orged at 2A00H). Prices are $19.50 and $24.50.

DTI General Ledger program.

DTI Payroll program.

A Block-Structured Language for Microcomputers

You think that there's no viable alternative to BASIC? You haven't heard about XPLO.

Larry Fish
123 E. Arkansas
Denver CO 80210

Ever since BASIC was introduced as a microcomputer language, there has been considerable discussion about an alternative high-level language. A number of languages have been suggested as alternatives; however, none of these languages is really suited to the needs of small systems. XPLO is the first viable alternative language for microcomputers.

Briefly, XPLO is a block-structured high-level language designed specifically for eight-bit microprocessors. It is five to ten times faster than the fastest BASIC. It is a compiled language and yet requires only 16K to 24K of memory, without a disk. But most important of all, it is available now.

Why Block Structured?

Block-structured languages were designed to solve problems that develop in conventional languages. If you have ever written a long BASIC program, you have probably encountered these problems.

1. *Complexity.* As a program grows in length, its complexity grows geometrically. In languages where any routine can call any other routine, programs tend to become complex webs of subroutine calls. Block structure solves this problem by organizing the program into clean, logical blocks. As a block-structured program grows, it becomes longer but not more complex.

Block structure helps the programmer deal with complexity in another way. The human mind can only grasp a certain amount of information at one time. The easiest way to deal with any program is to break it down into simple, easy-to-understand steps. Even a complex program can be written easily by breaking it down into small modules. Here again, block structure naturally organizes programs into small, easily understood blocks.

2. *Collision of Variables.* As a program grows in size, more and more variables are used to store data or carry information. With more variables, it becomes easy for the programmer to lose track of what each variable is doing in each subroutine. Eventually, variables collide and you find that your Star Trek program is going out to lunch because the variable that holds the Enterprise's shield power is being eaten by the Klingon navigation routine.

In block-structured languages, the programmer has complete control of each variable. Block-structured languages allow each variable to be defined either locally or globally. This means that variables can be set up to be active only inside certain routines or active for all routines.

To make things clearer, let's write a small program in XPLO. Because of the structure of the language, we can begin by describing the task in plain English. The program we will write is a simple guessing game in which the computer selects a number between 1 and 100 and we try to guess the number. After each guess, the program will tell us whether we are high or low. Here are the steps the program goes through:

1. Think of a number.
2. Get a guess from the keyboard.
3. Test the guess against the computer's number.
4. Do steps 2 and 3 until the guess is correct.

The steps translated into XPLO are shown in Program A. Notice that the program is almost word-for-word the same as the step-by-step description of the task. First we MAKEANUMBER, and WHILE the GUESSES are INCORRECT we INPUT a GUESS and TEST the GUESS. BEGINs and ENDs are used to divide the program up into logical blocks. This part of the program has two logical blocks, one inside the other.

Obviously, there must be more to this program, since XPLO doesn't yet know how to make a number, input a guess or test the guess. Each of these operations is a *subroutine* to the main program. In XPLO, subroutines are called procedures. We are now going to

```
'BEGIN'
MAKEANUMBER;
'WHILE' GUESS=INCORRECT 'DO'
                  'BEGIN'
                           INPUTGUESS;
                           TESTGUESS;
                  'END';
'END'
```

Program A.

```
'PROCEDURE'      MAKEANUMBER;
'BEGIN'
         NUMBER:=RANDOM(100);
'END';
```

Program B.

```
'PROCEDURE'      INPUTGUESS;
'BEGIN'
         GUESS:=INPUT(0);
'END';
```

Program C.

```
'PROCEDURE'      TESTGUESS;
'BEGIN'
         'IF' NUMBER=GUESS 'THEN'
         'BEGIN'
                  TEXT(0,"CORRECT!!");
                  TRY:=1;
         'END'
         'ELSE'
                  'IF' NUMBER<GUESS 'THEN'
                  TEXT(0,"TOO HIGH")
                  'ELSE' TEXT(0,"TOO LOW");
CRLF(0);
'END';
```

Program D.

write each procedure as a complete program block (see Program B).

This procedure generates a random number and puts that number in the variable NUMBER. Program C gets a number from input device number zero and stores it in the variable GUESS. In XPLO, as many as ten different input and output devices can be called directly from the program. This allows XPLO to read and write data directly to disks, printers, CRTs, etc.

Program D is a bit more complicated, but it is still easy to understand. If the computer's NUMBER is equal to our GUESS, then we execute one block of code; if they are not equal, then we execute another block. If the numbers are equal, we tell the user that the guess is correct; if they are not equal, we test if the guess is high or low and tell the user.

The Program

There are two new constructs in the final version of the program (see program listing). CODE allows the programmer to assign names to XPLO functions. For example, the word RANDOM calls the random number function. The programmer need only use those functions necessary to the task and can assign names that add clarity and readability to the program.

INTEGER assigns a name and memory space for each of the variables. Because of the way in which the variables have been set up in this program, each of the variables can be used by any procedure. If we had defined the variables inside a procedure block, the variables would have been active only within the procedure.

Block-structured programs can be thought of as a series of boxes. Each box has only one entrance and only one exit. The program enters at BEGIN and exits through END. Each box can contain sub-boxes, executable statements or calls to procedures (see Fig. 1).

Our program consists of four boxes: three subroutines and a main program box. Each block is a simple, complete operation. Programs are built a brick at a time from these elementary blocks. Even the most complicated programs, such as assemblers and compilers, can be constructed from simple blocks.

Notice that the main procedure is the last block in the program. Reading an XPLO program starts at the bottom to get the main sweep of the program and works up to the details in the subroutines.

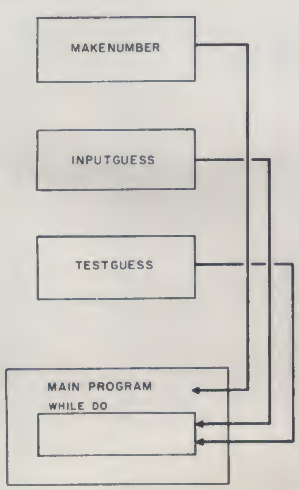


Fig. 1.

Now we can go back and fill in some of the details of the language. One of the most important qualities of a computer language is the way in which it deals with data. XPLO uses three techniques for efficiently dealing with data. These techniques are *scope, dynamic core allocation* and *parameter passing*.

Scope

Scope defines the area in which a variable is active. In many languages, the user has no control over the scope of a variable's activity. For example, in BASIC once a variable is created, it remains active for the entire program. In XPLO the scope of a variable is controlled by where the variable is defined. Variables are active only within their own block or within procedures nested inside that block.

In this way, some variables can be active in only one or two procedures, while others are active for all procedures. It is even possible to have several different variables with the same name and different areas of activity. If more than one variable with the same name is active, the most local varlable has precedence.

Dynamic Core Allocation

Dynamic core allocation is a logical extension of the idea of scope. Whenever XPLO completes the execution of a subroutine block, certain variables local to that block are no longer needed by the program. When a variable is no longer active, XPLO returns the unused space to the user's memory pool for use by other routines.

In contrast, variables created in BASIC take up memory space throughout a program's

```
'CODE' CRLF=9,RANDOM=1,INPUT=10,TEXT=12;

'INTEGER'   GUESS,NUMBER,INCORRECT,TRY;


'PROCEDURE' MAKEANUMBER;
'BEGIN'
    NUMBER:=RANDOM(100);
'END';


'PROCEDURE' INPUTGUESS;
'BEGIN'
    GUESS:=INPUT(0);
'END';


'PROCEDURE' TESTGUESS;
'BEGIN'
    'IF' NUMBER=GUESS 'THEN'
    'BEGIN'
            TEXT(0,"CORRECT!!");
            TRY:=1;
    'END'
    'ELSE'
            'IF' NUMBER<GUESS 'THEN'
            TEXT(0,"TOO HIGH")
            'ELSE' TEXT(0,"TOO LOW");
CRLF(0);
'END';


'BEGIN'
INCORRECT:=0;
TRY:=INCORRECT;
MAKEANUMBER;
'WHILE' TRY=INCORRECT 'DO'
                      'BEGIN'
                              TEXT(0,"GUESS: ");
                              INPUTGUESS;
                              TESTGUESS;
                      'END';
'END'
```

Program listing.

execution. The variable space in a BASIC program is always the sum total of all of the variables used in the program. XPLO programs use an absolute minimum of variable space.

Passing Parameters

Passing parameters is the way in which one routine communicates data to another. In many conventional languages, the data is sent from one routine to another by placing it in a variable and calling the routine. The programmer must know in advance which variable names are used by the receiving routine.

In XPLO, information being sent to another routine is simply tacked on to the end of the call. For example,

```
TEST(X,Y,Z);
```

calls a procedure named TEST and sends the variables X, Y and Z. When the call reaches the procedure, the values of X, Y and Z are placed into the first three variable names defined in that procedure. If the first three variables defined in TEST are A, B and C, then the value of X will be passed to A; Y will go into B and Z into C. This technique makes each subroutine a clean and completely independent operation.

Booleans

The Boolean operators AND, OR and NOT are available in XPLO. The Booleans are set up so that they operate upon individual bits. For example, the statement "X: = Y&4" indicates that Y is ANDed with the numerical value 4. Since 4 is the binary number 0100, this operation masks off all bits except bit three.

The ability to operate on individual bits gives XPLO the flexibility to link directly with machine-level operations. This enables the programmer to blast PROMs, read I/O ports and operate joysticks directly from the high-level language.

XPLO in XPLO

One of the most interesting things about XPLO is that the compiler is written in XPLO. This means that the compiler can compile itself and that new features can be added to the language by editing XPLO and compiling the new compiler. Thus each new version of the language is brought to life by the old version. Where did the first version of the compiler come from? The first version of the compiler was written in ALGOL.

Portability

The XPLO compiler translates the source program into an intermediate language called I2L. The I2L code is interpreted and executed in an I2L interpreter written in machine language. I2L is very close to machine language. It contains 28 op codes that are easily implemented in any machine language. Thus, XPLO can be run on any machine by writing the relatively simple interpreter for the particular CPU.

I2L interpreters run about 2K in length. Since all device-specific I/O is contained within the interpreter, the exact same compiler can run on all machines. Once an interpreter is written for a particular CPU, the user need only load the compiler to have the complete XPLO language running on his system.

XPLO is ideally suited to high-speed tasks such as real-time graphics. Arcade-type video games complete with sound effects are easily generated in XPLO. Assemblers, editors and operating systems can also be written in XPLO. XPLO could even be used to father a new language in the same way ALGOL fathered XPLO. A compiler for the new language could be written in XPLO and then each new compiler would be written in the new language.

Availability

XPLO was written specifically for microprocessor systems by Peter Boyle. At the present time, I2L interpreters exist for the 6502, Z-80 and the PDP-10.

A complete program development system is available for Apple II, KIM-1 and TIM systems. The basic package is $45 for KIM and TIM versions and $35 for the Apple II version. A detailed user's manual is also available for $17.25. Packages for other processors will be available in the near future. The packages operate in 16K to 32K of memory depending on the system. They include a compact, versatile editor that allows program creation and compilation to be entirely core resident.

The basic package consists of a memory image cassette or paper tape of the interpreter, compiler and editor. The user's manual is a detailed, step-by-step introduction to the language. It contains over 60 pages of concise description and example programs. It is clearly written so that even a novice can master XPLO.

The Apple II package contains a complete set of functions to drive the high and low resolution graphics, the game paddles and the speaker. A cross-referenced assembly listing for the interpreter and editor is available as a separate package.

KIM and TIM packages are available from The 6502 Program Exchange, 2920 Moana, Reno NV 89509. The Apple II package is available from P. J. R. Boyle, 1337 Adams, Denver CO 80220. ■

Reader Service—see page 179

✓ *Reader Service—see page 179*

KILOBAUD KLASSROOM NO. 16

Computer I/O IV

Input-output is not a FIFO (fast-in/fast-out) subject. Here's I/O installment number 4.

Peter A. Stark
PO Box 209
Mt. Kisco NY 10549

In October 1978 we started our discussion of computer input and output (I/O) circuits. To begin with, we looked at the external connections to a computer—serial and parallel ports, serial ASCII character coding, current loop and RS-232C connections and handshaking with a parallel port. In December we examined the port circuitry itself and saw how the port connected to the microprocessor.

This month we continue our discussion by looking at some specialized I/O ICs and also by describing the function of interrupts.

Introduction

As we broke off in December, I left you with a small homework problem—how would you connect something such as a seven-segment LED display to the processor? That happens to be an appropriate introduction to this month's topic.

Actually, the way the problem was worded left several unanswered questions: what kind of LED? . . . decoded? . . . how many digits? . . . latched? . . . *multiplexed?* Let's start by discussing the choices.

A single LED is just a gallium-arsenide-phosphide diode that happens to emit light when it is forward-biased. This requires a current of around 10-20 mA and produces a voltage drop across the diode of about 2 volts. You have already used such diodes in many experiments. The seven-segment LED readout is simply a combination of seven such LEDs in a special package, arranged in such a way that lighting up different combinations of the seven LEDs displays a number.

Fig. 1 shows a Fairchild FND 70 common-cathode LED. As shown in Fig. 1a, the seven LEDs are arranged in a figure-eight pattern, with each line called a segment. They are labeled a through g, as shown in Fig. 1b; most seven-segment readouts also include an eighth LED for use as a decimal point, or dp.

The FND 70 is housed in a 10-pin dual in-line package (DIP), with the pins identified as in Fig. 1c. To permit the use of just ten pins rather than 16, all of the diode cathodes are connected together. For this reason the FND 70 is called a common-cathode readout.

Fig. 2 shows the pin-out of another popular readout, the Monsanto MAN-1 or Litronix DL-10. As you can see, not only are the pins different, but also this time it is the anodes that are connected together; hence this is a common-anode readout.

Common-cathode and common-anode readouts are not interchangeable in a given circuit, since they must be connected and driven differently. With the common-cathode readout, the cathodes must be connected to a common negative or ground voltage, while the anodes are connected to a positive voltage through resistors. With the common-anode readout, the situation is reversed.

Fig. 3 shows two common ways of controlling the segments of common-cathode displays. In Fig. 3a, each segment is in series with a resistor, which determines how much current flows, and a transistor, which turns the current on or off. When a transistor is biased off, no current can flow through that LED segment, and it is therefore off. When the transistor is biased on, current flows through the LED to light it up.

Although NPN transistors are shown here, they could just as well be replaced by PNP transistors. Although there are no 7400-series TTL ICs designed to control the segments in this way, there is a 75491 MOS-to-LED quad segment driver IC that has been specifically designed for interfacing a MOS calculator IC to an LED display.

Fig. 3b shows a common way of controlling a common-cathode LED readout directly from TTL circuitry. This time, each LED gets power directly from the +5 volt line through a resistor. It is therefore lit up except when the shunt transistor is turned on; when this occurs, the transistor shorts the LED diode and shunts the diode current to ground. This prevents any current from reaching the diode. The 7448 BCD-to-seven-segment decoder that you used in Experiment #17 is used this way.

In contrast, the circuitry for a common-anode readout is simpler. As shown in Fig. 4, each segment is connected in series with a resistor and transistor. Turning the transistor on and off controls the LED. This is a common approach taken by the 7447 BCD-to-seven-segment decoder.

How would we connect such an LED to a computer output port? Any of the circuits of Figs. 3 or 4 could be connected to an output port, much like that of Fig. 4 in December's Klassroom. (Be careful to put 1k to 10k series resistors in series with the transistor bases to prevent burning them out!) The only difference in the circuits is that in Fig. 3b a high

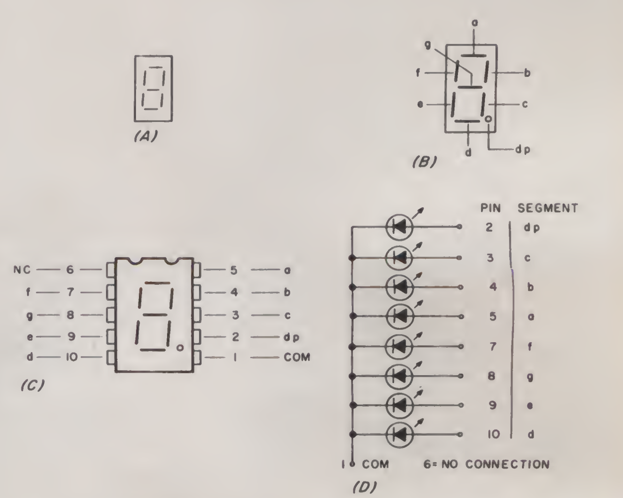


Fig. 1. A typical common-cathode seven-segment LED, the FND 70.

output from the port would turn a segment off, while in the other two circuits a high would turn a segment on. The computer program would have to take that into account.

Such a circuit would give the computer complete control of every segment. Not only could the digits 0 through 9 be lit up, but letters such as A, H or d would, too. On the other hand, we use up an entire port for just one display digit. (The only part of the port that we don't need is the handshaking part.) Moreover, the computer program must now be able to translate whatever digit it wants to display into the appropriate combination of segments.

There is another way to do the connection—through a 7447 or 7448 BCD-to-seven-segment decoder. You should recall that the 7448 had only four inputs, through which we entered a four-bit number in binary coded decimal (binary for the digits 0 through 9). The 7448 then in turn controlled the seven segments of an LED.

If we connected such a decoder between the output port and the LED, we would need only four bits from the port to control one LED. In this way, an eight-bit output port could provide two four-bit numbers to two 7448 decoders, and thus control two LEDs. This gives us more for each port but sacrifices the ability to display letters or other characters.

An Eight-Digit Readout

Let's assume that we would like to provide an eight-digit readout for our computer. This involves eight readouts for a total of 56 segments (64 if you also want to use the decimal points). Now we need eight ports for direct control, or four ports if we use separate decoders such as the 7447 or 7448.

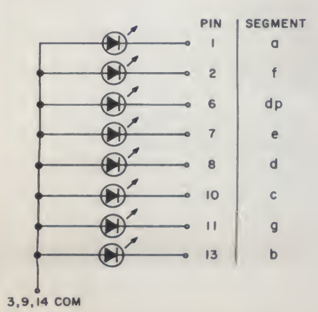


Fig. 2. Pin-out of the MAN-1 and DL-10.

Assuming that we want direct control over each segment, we now need eight address decoders, 64 flip-flops to latch each bit, 64 control transistors, 64 current-limiting resistors and several hundred connections. This is no longer an easy interface to wire! It's a problem that every calculator designer faces.

This is where a technique called *multiplexing* comes to the rescue. We have run across the word before—it means sharing a single data path for different pieces of data. In this case, we share two address decoders, 16 flip-flops, 16 transistors and only eight current-limiting resistors among all the displays.

It is done with time division multiplexing, that is, using the same circuitry for different digits at different times. Basically, the circuitry lights up only one of the eight digits at a time; each digit is displayed for a short time and then turned off, so that the next digit can be shown. Thus the digits are flickering on and off at a high rate, but this happens so fast that the human eye is fooled into thinking that all eight digits are on all the time.

Now we need two sets of transistors—one to control all the segments and the other to turn entire digits on and off. Fig. 5 shows a scheme for doing this for common-cathode LEDs such as the FND 70. Although only three readouts are shown because of lack of space, eight or even more can be used.

As Fig. 5 shows, all like-segment anodes of all the LED readouts are connected together and driven by one transistor. The eight transistors at the top then feed the seven segments and the decimal points. They are connected to one output port, which we call here Port A, through eight 4.7k base resistors. When the port outputs a positive level on a given bit, the corresponding transistor goes on and lights up its

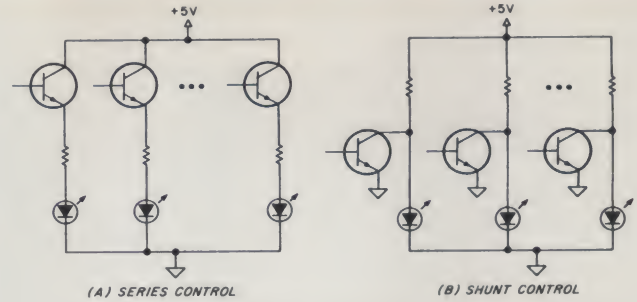


Fig. 3. Controlling common-cathode LEDs.

segment.

For best operation, each transistor should have its own 100 Ohm current-limiting resistor in series with the collector or emitter, but we've taken the shortcut of putting in just one in series with the +5 volt supply.

At the bottom, we see that the common cathodes of each LED display go to ground through another set of transistors. A given digit will go on only if its digit drive transistor is on; these transistors are controlled by a second output port, called Port B on the diagram. Each transistor goes on if the corresponding bit output from Port B goes high. If this port has eight output bits, then up to eight transistors (and digits) can be controlled by it, though only three are shown in Fig. 5.

To show how such a system would work, let's suppose that a 4 is to be displayed on LED 2. The computer would output a high to segments b, c, f and g on Port A and a high to the digit select transistor for LED 2 from Port B. Although all of the LEDs would be getting power for segments b, c, f and g, no current could flow through LEDs 1 and 3 since their cathodes are not grounded.

This system allows the computer to control up to eight LED displays through just two ports and to display all possible combinations of segments, not just the digits 0 through 9. In fact, this system can also display the hexadecimal digits A, b, C, d, E and F, although the b and d must be in lowercase and there can be some confusion between the b and number 6.

But to display an entire eight-

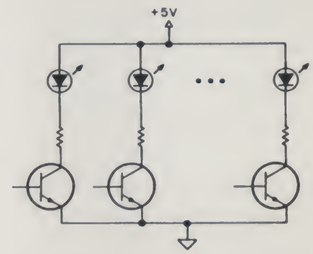


Fig. 4. Controlling common-anode LEDs.

digit number, the computer must constantly send out new data, one digit at a time, to refresh the display fast enough so that to the eye it will look continuously on. This can take up quite a lot of the processor's time, not just in deciding which segments to light and sending out the port data, but also in keeping tabs on when to turn off one digit and turn on the next.

With this as an introduction, we're finally ready to go on to more I/O circuitry.

Specialized I/O Chips

To drive the LED output interface of Fig. 5, we needed two output ports. Although in this case we didn't need any handshaking circuits, most other applications would need them as well. As we saw in the last Klassroom, the circuitry to provide two complete ports with handshaking can involve a substantial number of ICs. For this reason a number of specialized ICs provide this function on one chip.

Probably the simplest is Intel's 8212 8-Bit Input/Output Port, part of the 8080 family, shown in Fig. 6. (It is also available under the number 74S412.) As you can see, it has eight

data inputs to eight type-D flip-flops; the flip-flop outputs go to three-state buffers, which then feed the data outputs. Finally, there are some fairly simple control circuits.

This IC can be used as either an input port or an output port. For input, the 8212 inputs would come from an external device, and the data outputs would connect to the processor's data bus. For use as an output port, the 8212 inputs would connect to the data bus, and the outputs would feed the external device. The MD, or MoDe, pin determines which way the 8212 will be used.

Most of the 8212 pins are fairly simple to understand. When the $\overline{\text{CLR}}$ pin is grounded, all the flip-flops in the IC are cleared. This will most likely be done when the computer system is first turned on. The $\overline{\text{DS1}}$ and DS2 pins are device selects, similar to the chip selects on memory ICs. In order to use the 8212, the processor would have to make $\overline{\text{DS1}}$ low and DS2 high. In most 8080 systems, the $\overline{\text{DS1}}$ pin would go to the $\overline{\text{I/OR}}$ signal (or $\overline{\text{MEMR}}$ if memory-mapped I/O were being used), while DS2 would come from the port's address decoder.

The $\overline{\text{INT}}$ (INTerrupt) signal is an output back to the processor, which we'll talk about later on.

First, let's suppose that we want to use the 8212 as an input port; to do this, we ground the MD pin, which puts the 8212 into the input mode. The STB or STroBe pin would now go to the input device and probably be connected to its DATA READY line. Each time there is data to be sent to the computer, the device (card reader, tape reader or whatever) would put a pulse on the STB pin, which would read the data into the flip-flops. When this happens, the $\overline{\text{INT}}$ signal goes on (low, because of the bar over the INT), and this tells the processor that there is data waiting in the flip-flop latches.

When the processor is ready to read this data, it selects the 8212 by making $\overline{\text{DS1}}$ low and DS2 high. As a result, the 8212 enables the three-state buffers and sends the flip-flop data to the data bus. This action also turns off the INT signal; more on this later.

The operation of this IC is different when it is used as an output port. To do this, we connect the data inputs to the processor data bus, connect the MD mode pin to +5 volts and take the output to the output device from the eight data output pins. When the MD signal is high, the three-state buffers are always turned on, so that the flip-flop data is always being sent out to the data outputs.

Now when the processor wants to output data, it simply puts it on the data bus and selects the 8212 with the $\overline{\text{DS1}}$ and DS2 inputs. As soon as this happens, the data is latched into the flip-flops and sent to the outputs through the buffers.

The 8212 is only an example, and a simple one at that, of some of the sophisticated I/O chips available from microprocessor manufacturers. For instance, National Semiconductor has the DP8301 Microprocessor Interface Latch Element (MILE), which also provides a single I/O port, but its port can be bidirectional. Furthermore, the MILE has more extensive handshaking circuitry.

Another popular parallel I/O device is the MC6821 Peripheral Interface Adapter (PIA) from Motorola. This device was earlier available under the number MC6820 and is also made by MOS Technology under the number MCS6520. It not only provides full handshaking, but also has two ports in one package. This IC would be a perfect match for our eight-digit LED display.

Look at Fig. 7 for the block diagram of the PIA. On the left, we have the connections to the processor; on the right are the two I/O ports.

On the processor side, the eight data pins go straight to the data bus. R/$\overline{\text{W}}$, Ø2 and $\overline{\text{RESET}}$ go straight to the processor control signals. CS0, CS1 and $\overline{\text{CS2}}$ are three chip enable signals that have to be high, high and low, respectively, and go to address decoding circuits. Using three separate chip selects in this combination makes address decoding simpler; with incomplete address decoding these three may be able to do the entire job by looking at three bits. Even with full decoding, having three inputs makes it easier to split the decoding circuitry into several smaller sections.

$\overline{\text{IRQA}}$ and $\overline{\text{IRQB}}$ are Interrupt ReQuest pins that we'll discuss shortly.

That leaves RS0 and RS1; they determine whether data bus information is data or control signals and whether it applies to Port A or Port B. In actual use, they usually connect to the two least significant bits of the address bus. Usually the 14 most significant bits are decoded by the decoders, and the last two go to RS0 and RS1 in such a way that there are four addresses for the PIA, depending on the state of these two bits. (This is why we mentioned in our last article that the SWTP computer used four addresses per I/O device.)

The MC6821 is actually quite a complex IC. Not only does it have two separate ports, but within each port some bits can be used for output while others are for input. As a result, we could split the ports up to give 15 output bits and one input, or any combination.

There are also various options for the handshaking lines, which make the PIA suitable for hundreds of different jobs. Early Motorola and SWTP systems even used the PIA for serial data I/O. Moreover, these functions can be changed during execution of a program, since the PIA is programmable. This means that control information fed from the processor to the PIA over the data bus selects the operating mode of the device. This is why four addresses are needed for the two ports—two are for data and the other two for control.

Since you are probably not going to use the 6821, it doesn't pay to go into great detail about it. Let's just say that such an IC could, with just a bit of extra address decoding, make an excellent driver for the LED display

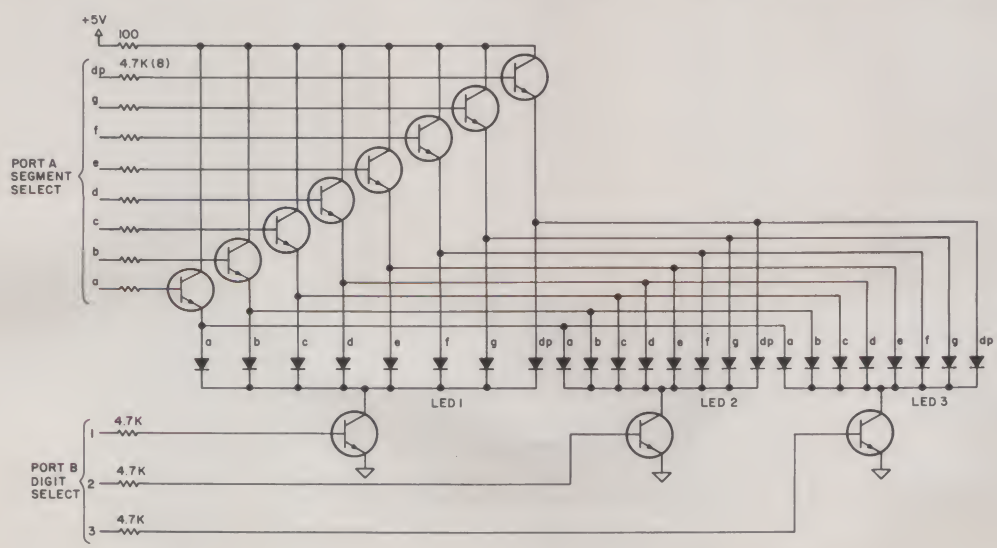


Fig. 5. Connecting several LED readouts to two output ports.

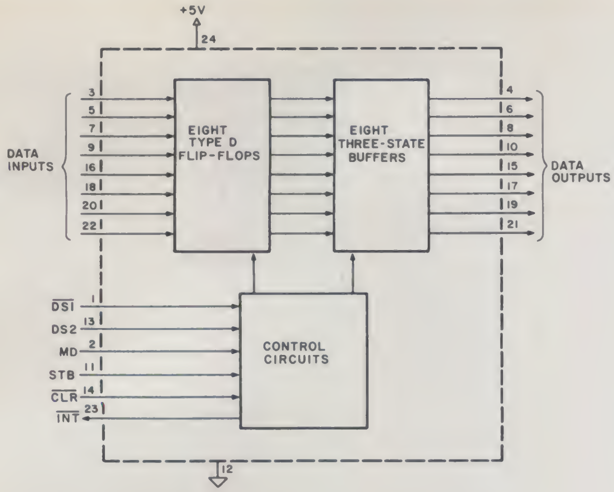


Fig. 6. The 8212 8-Bit Input/Output Port.

we started with in this installment.

Most microprocessor manufacturers make their own versions of the PIA. Intel's equivalent is the 8255 Programmable Peripheral Interface (PPI); Intersil makes the IM6101 Parallel Interface Element (PIE); and Zilog makes the Z-80 Parallel I/O Interface (PIO). It is a useful device for interfacing everything from keyboards to printers.

Interrupts

Twice so far we have mentioned the idea of interrupts, in the form of INT and IRQ signals that come out of some of the parallel port ICs and go to the processor. What are they?

An interrupt is simply a signal generated by an I/O device or its interface that goes back to the processor and interrupts its normal operation. It forces the microprocessor to pause in its normal work and pay attention to the I/O device.

To give you some idea of the importance of interrupts, let's talk about two subjects we have already covered.

In Klassroom No. 15, we discussed an interface for a paper tape reader. We developed a memory-mapped circuit that used address 8004 for data and address 8005 for handshaking control signals. Specifically, we used address 8005 to test whether a character had been read by the reader. We gave a typical program that went like this:

STEP 1: INPUT FROM ADDRESS 8005
STEP 2: IF NOT READY, RETURN TO STEP 1
STEP 3: INPUT DATA FROM ADDRESS 8004
STEP 4: WRITE TO ADDRESS 8005
STEP 5: CONTINUE

We mentioned that steps 1 and 2 comprised a wait loop, and that the processor would simply repeat those two steps until a character had been read.

If the tape reader is slow, there might be a fairly long wait. In fact, during the reading of a long tape it is entirely possible that the microprocessor spends 99 percent of its time waiting. Wouldn't it be nice if this time could be used productively, rather than being wasted?

In simple systems, we could make use of this time simply by changing step 2 to go do something else and return sometime later. But the problem remains —how long can you do something else before coming back? If you return too soon, you waste time. If you wait too long, you may miss the next tape character altogether. What we really need is some way to go off, do something else and leave it up to the tape reader to call the processor when it is finally ready. This is exactly what is accomplished by an interrupt.

Let's think about another example. At the beginning of this month's installment, we developed an LED output display that drove up to eight seven-segment LED readouts with just two output ports. But we mentioned that the multiplexing technique we were using required us to turn the LEDs off and on very fast so that to the eye they would all appear continuously on, even though only one was on at any one time.

A good speed for this is each LED display flashing on at least 30 or 40 times per second. With eight digits, we must switch LEDs at least 240-320 times per second to avoid visible flicker. A repetition rate of 500 or 1000 per second would be somewhat better. Now comes an interesting question—how does the processor know when it's time to switch?

The most obvious way is to provide a wait loop in a program, which provides a one or two millisecond delay. Every millisecond or two the program would finish the loop and go on to light up the next display. This is a perfectly good way to do it and has been used in many systems. But again, the disadvantage is that while all this is going on, the microprocessor can't do anything else. Although it could jump out of the loop to do something else, there is no guarantee that it would return back in time.

The problem is again easily fixed by using an interrupt system. In this case, we build an external oscillator that provides a pulse every millisecond or two to the processor's interrupt input. Regardless of what the processor is doing, this external input will stop it and cause it to update (refresh) the LED display. This oscillator is called a real-time clock, since it synchronizes the processor to the real time. It's exactly what the Heath H8 computer does to make sure that its LED display is fully functional even when the computer is working on some other program.

While we're at it, we can use the real-time clock oscillator to keep track of the actual time. Each time the clock interrupts the processor, we add one to an internal counter. At any time, the count in that counter tells how many clock pulses occurred since the counting started. If the oscillator frequency is precisely controlled, the computer can calculate the exact time from this number of counts. This is how many computers tell time and provide the time and date printouts you often see on program runs.

But back to interrupts. In its simplest form, the interrupt

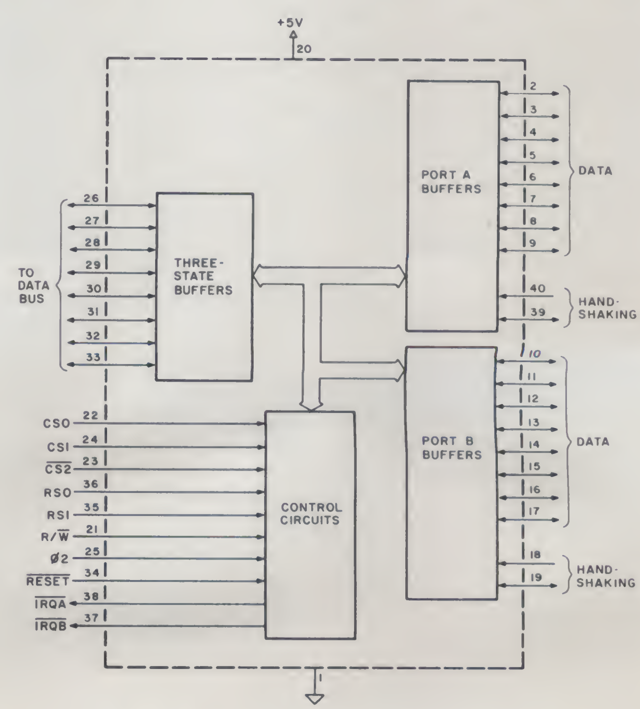


Fig. 7. The MC6821 Peripheral Interface Adapter.

system simply consists of an extra microprocessor pin labeled INT, IRQ or something like that. INT, of course, stands for INTerrupt, while IRQ stands for Interrupt ReQuest. When some external device puts a signal on this pin (called an interrupt request), the processor stops whatever it's doing and instead goes to perform some other program. (At this point we say that the interrupt has been granted.)

This new program is usually called an Interrupt Service Subroutine (ISS). When the ISS program is done, the processor will return to the job it was doing when it was interrupted. But most interrupt systems are a little more complicated.

First of all, there must be some way for the processor to prevent interrupts from occurring whenever some very sensitive work is being done that should not be interrupted. This is usually handled with an internal Interrupt Enable or Interrupt Mask flip-flop, which can be controlled by the program and whose output is gated with the interrupt request signal to prevent it from getting through at certain times.

Second, it is important for the processor to know where it was and what it was doing when it was interrupted, so that it can pick up where it left off when the interrupt is done. This information is called program status and has to be saved somewhere. Information as to where the main program was when interrupted is generally saved automatically by the microprocessor in the same way as subroutine return addresses are; other program status data may also be saved by the microprocessor hardware or may have to be handled by software as part of the ISS program.

Third, in a reasonably sized system, an interrupt may be caused by different events. It's necessary to give the processor some way of finding out what caused it so it can take the proper action.

Interrupts in the 6800

The Motorola 6800 microprocessor has one of the simpler interrupt systems around, so let's spend some time looking at it. Fig. 8 shows a simplified diagram of the interrupt inputs.

There are two interrupt pins, labeled $\overline{\mathrm{IRQ}}$ (Interrupt ReQuest) and $\overline{\mathrm{NMI}}$ (Non-Maskable Interrupt). Also used is one of the bits in an internal register called the Condition Code (CC) register, which is ANDed with the $\overline{\mathrm{IRQ}}$ input as shown.

The $\overline{\mathrm{IRQ}}$ input can only work when the I bit in the CC register is a 0, since only then will the low level on the $\overline{\mathrm{IRQ}}$ pin and the

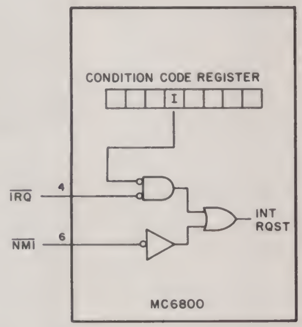


Fig. 8. Interrupt masking in the 6800 microprocessor.

low level from the I bit open up the AND gate to generate an internal interrupt request signal. Hence the $\overline{\mathrm{IRQ}}$ interrupt can be turned off (i.e., masked) by making the I bit in the CC register a 1. This is how a program can prevent interrupts from occurring at certain times.

On the other hand, the $\overline{\mathrm{NMI}}$ input goes straight through to the internal interrupt request line. Whenever it is grounded, an interrupt request is generated internally and the processor immediately interrupts. To avoid the processor interrupting over and over if this pin is held grounded too long, the $\overline{\mathrm{NMI}}$ pin responds only to the falling edge of an input. Thus only one interrupt will occur even if the $\overline{\mathrm{NMI}}$ pin stays low for a long time.

The reason for these two interrupt inputs is that they are intended for different purposes. The $\overline{\mathrm{IRQ}}$ input is the one normally used for I/O equipment; the $\overline{\mathrm{NMI}}$ input is intended for special or emergency situations that require an interrupt even though the processor may have masked the $\overline{\mathrm{IRQ}}$ input.

One example is a power-fail circuit that detects a falling power supply voltage. In this case, an $\overline{\mathrm{NMI}}$ interrupt might be generated to stop the processor and force it into an emergency ISS, which will store all data onto disk or other auxiliary memory in time to save all data before the power goes completely off. Even though the processor may be in the midst of an important task and doesn't want interrupts, such a power-fail interrupt may be more important still and has to get through.

When an interrupt occurs, the processor stores the program status—the contents of all internal registers—in memory in an area called the stack. Next, it turns on the I bit in the CC register so that any other $\overline{\mathrm{IRQ}}$ interrupts will not be able to get through to cause further interrupts. Finally, it looks in the top eight locations of memory for the address of the ISS to which it should go in response to the interrupt.

Within the top eight locations (which are usually ROM memory), there are four stored addresses; one of these is for the IRQ interrupt and another is for the NMI interrupt.

For example, in the SWTBUG ROM, which is supplied with the SWTP 6800 computer system, the top eight memory locations have the following four addresses:

Locations	Address	Used for
FFF8-FFF9	E000	IRQ
FFFA-FFFB	E18B	SWI
FFFC-FFFD	E1A7	NMI
FFFE-FFFF	E0D0	RESET

Two of these addresses are for the normal $\overline{\mathrm{IRQ}}$ and $\overline{\mathrm{NMI}}$ interrupts, one for doing a reset and one for an instruction called SWI, or SoftWare Interrupt. (These addresses were picked by the designers of this particular computer and are not part of the 6800 microprocessor. Other computers using the same 6800 IC would generally use different addresses.)

When an interrupt occurs, the processor will pick whichever address applies to the source of the interrupt and go to that address for the beginning of the ISS.

For instance, on an $\overline{\mathrm{IRQ}}$ interrupt the 6800 will go to an ISS at location E000; on an $\overline{\mathrm{NMI}}$ interrupt it will go to another ISS at location E1A7. This means that there can be different ISS programs, one for each kind of interrupt. In this way the processor can get to the required program faster than if it had to go to just one ISS and then determine, by program, which input caused it.

A group of addresses such as this, placed together and used by the processor as a table in which it "looks up" an address as it needs it, is often called a "jump table" or "transfer vector." In this case, we say that the different interrupts are "vectored" to different ISS routines, depending on their origin.

The idea behind vectoring interrupts is that the processor will be able to respond faster to an Interrupt if it does not have to go through a long routine first to find where the interrupt came from.

Actually, the interrupt system in the 6800 processor is not really vectored, despite everything we have said so far.

In most cases, there will be more than one interrupt source. In addition to a real-time clock, there might also be interrupts coming from each of the I/O devices. There might be anywhere from one to as many as ten or 20 devices in the system that could ask for an interrupt. But all of the interrupt sources (except for one) will be ORed together into the $\overline{\mathrm{IRQ}}$ pin and will therefore cause one particular ISS to be run. The only exception might be the power-fail interrupt, which would be all by itself on the NMI input. So the ISS would still have to go through the effort of finding out what really caused the interrupt in the first place. More on vectoring shortly.

In the general case, we could list the things an ISS would have to do:

1. Mask further interrupts so the ISS cannot get interrupted by something else. (This is almost always done by the processor hardware, rather than

by the ISS program.)

2. Save the status of the interrupted program so that it can later be continued without any loss of data. (In the 6800 this is also done by the hardware.)

3. Find out what caused the interrupt.

4. Run the appropriate program to take care of that interrupt.

5. Possibly check whether something else is going on that is also asking for an interrupt and return to step 3 if there is.

6. Unmask the interrupt system, get the status of the interrupted program back and return back to doing it.

Step 3 is needed because of the lack of sufficient hardware vectoring. Finding out what caused the interrupt is simple but possibly lengthy. It's done simply by checking all the DATA READY, DATA ACCEPTED and other handshaking lines to all the I/O devices to find one which indicates that some device needs help. This is called polling the device status bits.

Polling I/O devices to find the source of an interrupt is a perfectly acceptable idea in a smaller computer system or in one that does not operate at great speed. The problem is that in any sizable system that has programming complicated enough that several I/O devices are used at the same time, interrupts may be coming so close together, from different sources, that a substantial amount of time is taken up by the ISS programs looking to see where they came from. This can become especially troublesome if some of the I/O devices require fast response.

To get an idea of the problem, let's talk about a typical case: A small business system has several keyboard/CRT terminals used by order clerks, as well as a high-speed tape drive that records these orders. Let's assume that the approach is to accept orders from the terminals, do some simple checking on them to make sure there are no errors and store the data in the computer memory until an entire order for a given customer is finished. Then the computer turns on the tape and, at high speed, writes the entire order on tape.

Let's assume that a typical order consists of 1000 characters of data and that the tape operates at 10,000 characters per second; this works out to 100 microseconds (us) per character. Then the order will take 1/10 second to be written to tape. This is a short time but still long enough so that if the computer were to devote itself completely to the tape for that tenth of a second, it might miss a few characters coming in from the terminals.

Hence the computer cannot be completely tied up with the tape drive. Instead, it will have to be programmed so that the tape drive will interrupt the processor once every 100 us and ask for a character. Let's assume that for every tape interrupt, the processor needs 20 us just for general overhead—saving and restoring program status and so on—another 40 us to figure out where the interrupt came from and another 20 to send out the next character to the tape. The entire procedure then takes 80 us, so the processor will always be tied up 80 us, free for 20, tied up for 80 and so on.

OK, but now let's remember that there is a batch of CRT terminals connected here too. These will also generate an interrupt at times. Let's assume that the same thing exists here —20 us for overhead, 40 us to find out where the interrupt comes from and another 20 us to take care of it, for a total of 80.

The first thing you notice is that there isn't enough time between tape interrupts to take care of a terminal interrupt; you only have 20 us of free time between characters and need 80.

Although this sounds like quite a problem, there are three fairly simple ways to get around it. At least they are simple once you see what they are. The first, of course, is hardware vectoring to a specific ISS for each interrupt. In this way it may be possible to cut the total response time from 80 us to perhaps 40 or 50. (There may still be some additional overhead involved with the vectoring, so we may not be able to get all the way down to 40.) This means we will have to add some more hardware to the processor. Even so, it may not be good enough, because if we cut the time down to 50 us or just under, we will have just barely enough time to do both a tape and a terminal interrupt in the space of just 100 us.

Another idea is to forget about vectoring but set up some sort of a priority scheme so that the handshaking lines of fast I/O equipment will be examined first. In this way these devices will get faster service from the processor. By rewriting the ISS, we may be able to check the flags in such a way that it only takes 10 us to identify a tape interrupt, while it still takes 40 us to find the terminal interrupt. This doesn't help us much here, but it does at least solve the problem of what happens if two interrupts just happen to come at the same time—the tape interrupt will be found and taken care of first.

There is a third solution—set up the system so that the tape can interrupt the processor even if it is already busy with a CRT terminal interrupt. But watch out—don't ever let the CRT terminal interrupt the processor while it is taking care of the tape or you will lose a character. This again brings up the idea of a priority—more important devices (such as tape) can interrupt less important devices (such as terminals).

Although the 6800 does not provide interrupt vectoring for normal I/O interrupts, vectoring can be provided by the addition of some external hardware. In most larger computer systems all three of the above tricks are used: hardware vectoring, a priority scheme and allowing interrupts to interrupt each other.

The way it then usually works out is that there is more than one ISS routine. In some cases, there may be one ISS routine for each different I/O device; in other cases several devices may share the same ISS. In any case, there is the same priority scheme so that important interrupts—those from devices that need fast response—get taken care of first, and faster devices can interrupt slower ones.

The interrupts are then divided into levels: a level 7 interrupt is taken care of first, followed by level 6, etc. For instance, fast I/O devices such as disks or tapes might be on level 7; slow devices such as terminals might be on level 1; and console switches (such as a STOP switch) might be on level 0. The latter would only be taken care of when everything else has been done. (On the 6800, level 7 is considered the highest priority and level 0 the lowest; on other computer systems it's often reversed with level 0 the highest and 7 the lowest.)

This is actually an interesting concept. Many larger computers have a STOP button on the front panel, which can be used to stop their operation. But this button does not really connect to the control circuitry. This is because accidentally stopping the computer while it is in the middle of a program can cause a lot of grief.

To prevent this from happening, the STOP button merely generates an interrupt on the lowest level. It is then up to the interrupt servicing programs, which are part of the software called the operating system or monitor, to decide whether it is really safe to stop or not. For instance, in our example above it would not be advisable to stop the whole computer while it was writing on tape, since there would then be some incomplete information on the tape.

The Background Program

Let's look at our order entry example some more. Since an order is written on tape only after it has been completely entered from a CRT terminal, we can see that with just a few terminals there may be many times when the computer is just sitting there waiting for data to arrive from the terminals, with nothing else to do. Why not put this time to good use?

Let's think of something else

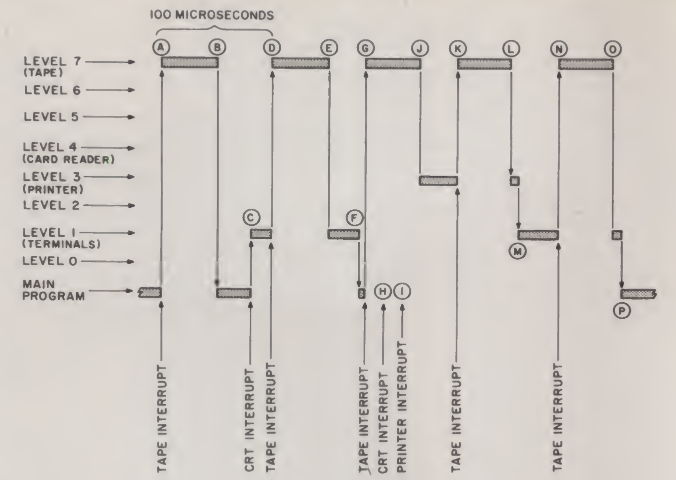


Fig. 9. A timing diagram showing interrupts interrupting each other.

for the computer to do in its free time. How about a program to read employee data and payroll information from punched cards and print paychecks? Since we already have the tape on level 7 and the terminals on level 1, let's connect the card reader to level 4 and the printer to level 3. Finally, the payroll program (which isn't very important) will be classified even less important than level 0.

Now we have everything ordered like this:

Level 7	Tape
Level 6	
Level 5	
Level 4	Card reader
Level 3	Printer
Level 2	
Level 1	CRT terminals
Level 0	
Last:	Payroll program

This shows us the most important actions at the top and the least important at the bottom.

Notice that we now appear to have two tasks going on at the same time: the order entry task and the payroll task.

The order entry would normally be called "interactive" because it interacts with the order clerks. On the other hand, the payroll program runs all by itself and is given to the computer as a batch of program cards, so it would be called "batch." Another pair of catchwords often used by data processing people is "foreground" and "background." The payroll program, which runs all the time that the computer isn't needed for something more important, would be the background program. The order taking, which takes over when data comes from the terminals, is the foreground (because it gets in front of the background).

Multiprogramming

Now we have a fairly complex process, with interrupts interrupting the background program and interrupting each other. This can be quite complicated, but a timing diagram such as Fig. 9 can help us see what is going on.

Refer to the circled letters in Fig. 9 as you read this description. Let's assume that the main (background) program is initially running and that a tape interrupt comes first at A:

A. Tape interrupt interrupts main program.
B. Tape ISS finishes, and main program continues.
C. CRT terminal interrupt interrupts main program.
D. Tape interrupt interrupts CRT ISS (100 us after the previous tape interrupt).
E. Tape ISS finishes, and CRT ISS continues.
F. CRT ISS finishes, and main program continues.
G. Tape interrupt again stops main program.
H. CRT interrupt comes but can't interrupt tape, so nothing happens.
I. Printer interrupt comes, but still nothing happens.
J. Tape ISS finally finishes, so the next highest level interrupt takes over and printer ISS starts.
K. Printer ISS isn't done yet, but tape interrupt starts the tape ISS.
L. Tape ISS finishes, so printer ISS continues.
M. Printer ISS finally finishes, so CRT terminal ISS can start.
N. Tape interrupts CRT terminal ISS.
O. Terminal ISS resumes when the tape is done.
P. Main program finally continues at end of terminal ISS.

As you can see, when a pair of background-foreground programs runs at the same time (which is called multi-programming), things can get fairly complicated. Both the hardware and the software have to be well designed to prevent things from going wrong.

Setting Up a Priority

Let's get back to that priority idea. Remember that this is a two-sided issue. We want a priority set up so that more important devices will be identified first, but we also want it so that a more important interrupt will be accepted while the processor is busy taking care of a less important one. The priority can be handled either by software or by hardware; let's look at the software approach first.

Remember that when an interrupt is granted and the processor jumps to an ISS, it turns off (masks) the interrupt system so that no further interrupts can occur. If the tape interrupt service routine doesn't turn off the mask, then it simply cannot be interrupted by a terminal. But if the CRT terminal's interrupt service routine does turn off the mask, then the tape interrupt will interrupt it.

In this way we set up a priority, but we still have to be careful that the tape interrupt doesn't break into the terminal ISS at times when something crucial is going on. In most ISS programs, there will still be some times when the interrupt is masked to avoid critical timing or other important parts being interfered with.

Another way to take care of the problem is with hardware. If two interrupt requests occur at the same time, the hardware will only let the tape interrupt go through. Once the processor is busy with a tape interrupt, the hardware will simply not let other interrupts get to the IRQ pin; it will hold them off until the tape interrupt service is completed.

The MC6828 PIC

As we mentioned before, the basic 6800 does not have any hardware for setting up priorities or for vectoring several interrupts. Although this can be done with software to some extent, Motorola has recognized

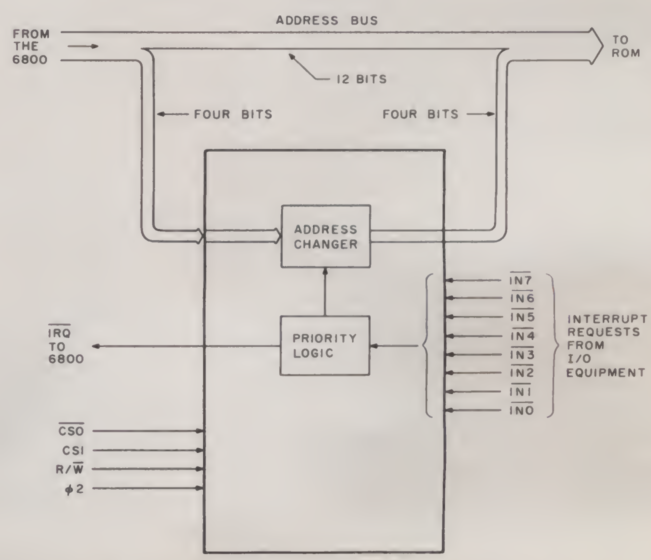


Fig. 10. The MC6828 Priority Interrupt Controller (PIC).

the need for a hardware solution and developed a special IC just for this purpose: the MC6828 Priority Interrupt Controller (PIC) shown in Fig. 10.

The MC6828 has eight interrupt inputs called $\overline{IN0}$ through $\overline{IN7}$, which go into its priority logic. When an interrupt request arrives on these inputs from an I/O device, if it is on a high enough level the MC6828 sends it on to the 6800 on the $\overline{IRQ}$ line. As you remember, the 6800 responds to an $\overline{IRQ}$ interrupt by going to locations FFF8-FFF9 in ROM for a vectored address of an ISS to jump to. But this read from locations FFF8 and FFF9 goes on the address bus.

As you can see, four bits of the address bus have to go through the MC6828 to get to the ROM. The 6828 recognizes this address and changes it. It sends to the ROM a different address, which depends on which interrupt request it received. In other words, the 6828 does its own vectoring and chooses one of eight new addresses, one for each possible interrupt source.

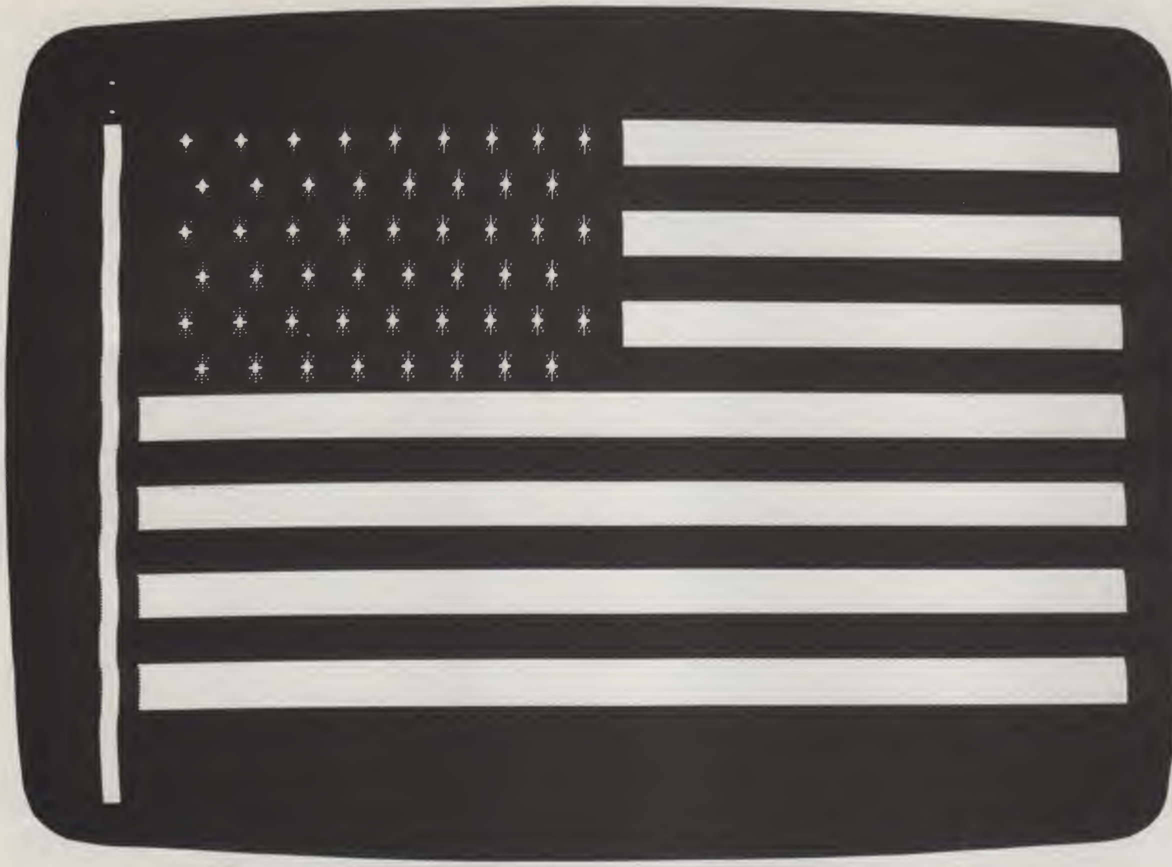
For example, on a level 0 interrupt the 6828 changes the addresses from FFF8-FFF9 to FFE8-FFE9; on a level 1 interrupt it changes the addresses from FFF8-FFF9 to FFEA-FFEB, and so on. This allows the programmer to put a completely different set of transfer vectors into ROM and split up the one $\overline{IRQ}$ interrupt into up to eight separate $\overline{IN}$ interrupts, each with its own vector.

How does the 6828 mask out lower-level interrupts? There is a way for the 6800 to tell the 6828 which level interrupt it is currently working on: by writing to a certain memory address. When this is done, the 6828 simply screens all incoming interrupts and prevents lower-level interrupts from interrupting higher level ones. All in all, it's a simple solution to a complex problem.

The other inputs to the 6828 are fairly obvious; there are chip selects $\overline{CS0}$ and CS1, which go to address decoders, an R/$\overline{W}$ signal that tells the 6828 whether the 6800 is reading or writing and, of course, a Ø2 clock signal.

Conclusion

We have seen all the things that an interrupt system needs to work and have looked at how the 6800 microprocessor does them. Next time we will see how the 8080 and Z-80 processors handle interrupts and will look at still another I/O method, the direct memory access, or DMA. See you then! ■



Michael A. Feldman
Faye A. Dion
34 Little River Lane
Middletown CT 06457

Computerized Climate Control

When Jack Frost started to nip and their wintertime heating bills began to soar, these two New Englanders devised a practical solution: they put an Elf in the thermostat.

Here's an application many of us have thought and talked about but very few have actually implemented . . . much less, sat down and wrote about. We think you'll enjoy, and appreciate, Mike and Faye's efforts. Those of you who tackle this project using a different micro as the controller should also generate an article and share your frustrations and rewards with the rest of us. By the way, Faye and Mike mentioned that there isn't a BASIC available for the 1802. Not quite true. Tom Pittman, Itty Bitty Computers, PO Box 23189, San Jose CA 95153, offers a Tiny BASIC for the 1802 (available only in paper tape—$5).—Editors.

For two programmers living in the same house, the idea of a personal computer for amusement as well as for more practical household applications, such as finance control, is, quite naturally, an appealing one.

But after returning home from work once too often to discover that one of us had forgotten to turn down one or more of the eight thermostats controlling the various zones of our electric heating system, we found the idea of computer-controlled heat downright compelling. As temperatures plummeted and our electric bill soared, we were delighted to discover, for our purposes, the Elf-II computer, marketed by Netronics, Inc.

Built around RCA's COSMAC microprocessor, the Elf has 256 bytes RAM, a hex keyboard and two seven-segment LEDs. It is a minimal system, but quite appropriate for our single-task application.

Our search for a small home computer began with the marketing thrust from manufacturers of hardware in the $500 to $1500 range. The systems were tempting, but too powerful for our simple control requirements. Most had BASIC, cassette interfaces, minimal operating systems and room for expansion. Of course, we could have become full-fledged hobbyists with one of these big micros, but since single-tasking is the rule in personal computing, we probably would have frozen as one or the other zapped Klingons or hunted the Wumpus.

Next we investigated single-board systems such as the Imsai controller, the KIM-1 and the Intercept Jr. All were attractive, but still too powerful (the KIM comes with I/O ports, ROM monitor, cassette and keyboard interfaces and 1K RAM, for example), and still too expensive ($250 to $300 range).

The COSMAC chip is beginning to receive attention in hobby circles. It now has a BASIC—Tiny—and it boasts an instruction set that is ideal for writing control programs. It uses an 8-bit byte (remember we have eight thermostats in our house), and can address up to 65K like most of the chips on the mar-

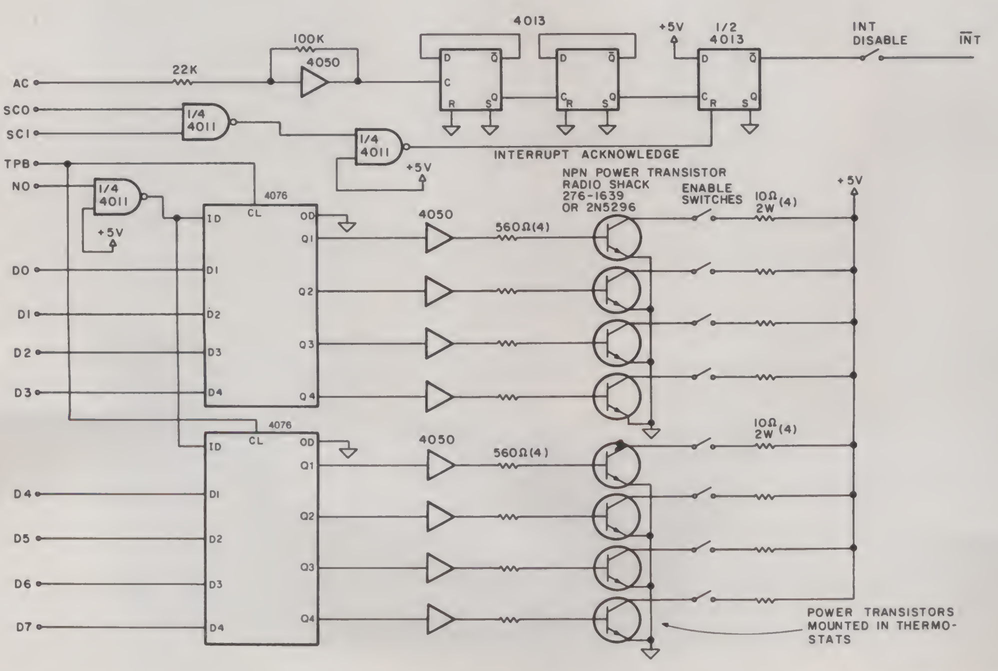
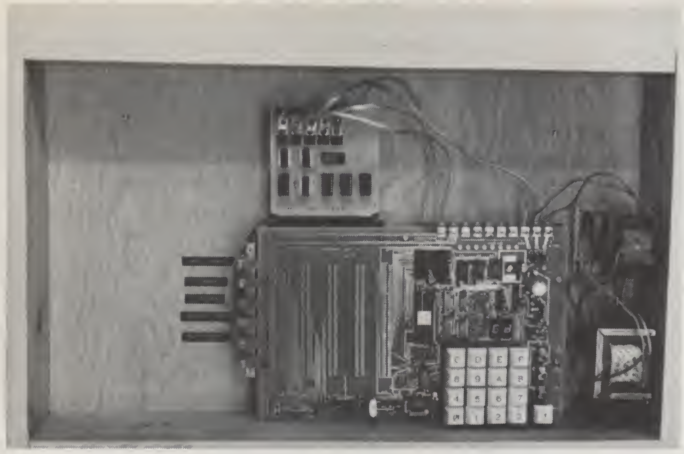


Fig. 1. The heat-control circuit. The top portion generates interrupts from the 60 Hz ac line, while the bottom part of the circuit acts as a simple output port interfaced to some power resistors.



The circuitry needed to control the heat fits on a small PC board. The board was assembled using wire-wrap techniques.

```
ADDR  CODE   LABEL     OPER,  OPERAND       COMMENT
00    F8 00            LDI    X'00'         R8 contains day
02    A8               PLO    8             of week
03    F8 00            LDI    X'00'         RG contains time of
05    A6               PLO    6             day in 6 min. units
06    F8 0A            LDI    A(START)      R3 Is the program
08    A3               PLO    3             counter
09    D3               SEP    3             -
0A    F8 FF  START     LDI    X'FF'         R2 Is stack ptr
0C    A2               PLO    2             -
0D    E7               SEX    7             Make R7 X register
0E    F8 51            LDI    A(INT)        R1 has Interrupt rtn
10    A1               PLO    1             address
11    F8 0F            LDI    X'0F'         R4 counts number of
13    A4               PLO    4             interrupts
14    F8 68            LDI    X'68'         Hex 168 sec/6 minutes
16    A5               PLO    5             counted In R5
17    F8 01            LDI    X'01'         -
19    B5               PHI    5             -
             .
             .         MAIN PROGRAM LOOP
             .
1A    F8 80  TOP       LDI    A(TABLE)      Put address of table
1C    A7               PLO    7             into X register
1D    86               GLO    6             D = current time
1E    F7     LOOP      SM                   D = D - M(7)
1F    32 2B            BZ     HIT           If zero, then table match
21    17               INC    7             Bump to next entry
22    17               INC    7             -
23    17               INC    7             -
24    F8 FF            LDI    X'FF'         Are we at end of table
26    F7               SM                   -
27    32 1A            BZ     TOP           Yes, start again
29    30 1D            BR     LOOP          No, try this entry
             .
             .         TABLE MATCHES, OUTPUT BYTE
             .
2B    17     HIT       INC    7             Point to weekday byte
2C    88               GLO    8             Get day of week
2D    FD04             SDI    X'04'         4-day of week
2F    33 32            BPZ    WEEKDAY       Negative means weekend
31    17               INC    7             Point at weekend byte
32    61     WEEKDAY   OUT    61            Output the byte
33    F8 3E            LDI    A(TIMESAVE)   Save current time
35    A7               PLO    7             -
36    86               GLO    6             Get time
37    57               STR    7             Store it
38    86     LOOP2     GLO    6             Get time
39    F7               SM                   Equal to saved time
3A    3A 1A            BNZ    TOP           No. Start again
3C    30 38            BR     LOOP2         Try again
3E    00     TIMESAVE  DS     X
             .
             .         INTERRUPT SERVICE ROUTINE
             .
4F    42     EXIT      LDA    2             Restore D reg
50    70               RET                  Return to mainline
51    22     INT       DEC    2             Int rtn starts here
52    78               SAV                  Save P,T regs
53    22               DEC    2             Decrement stack ptr
54    52               STR    2             Save D register
55    7A               REQ                  Turn off Q light
56    24               DEC    4             R4 = R4 - 1
57    84               GLO    4             D = R4
58    3A 4F            BNZ    EXIT          If not zero, return
5A    7B               SEQ                  Turn on Q light
5B    F8 0F            LDI    X'0F'         R4 = 15 (Ints/sec)
5D    A4               PLO    4             -
5E    25               DEC    5             R5 = R5 - 1
5F    85               GLO    5             D = R5
60    3A 4F            BNZ    EXIT          If not zero, return
62    95               GHI    5             Check high order byte
63    3A 4F            BNZ    EXIT          If not zero, return
65    F8 68            LDI    X'68'         Reset R5 to X'168'
67    A5               PLO    5             -
68    F8 01            LDI    X'01'         -
6A    B5               PHI    5             -
6B    22               DEC    2             Decrement stack ptr
6C    16               INC    6             Increment time
6D    86               GLO    6             Display time on LEDs
6E    52               STR    2             -
6F    64               OUT    4             -
70    FF F0            SMI    X'F0'         Does time = X'F0'
72    3A 4F            BNZ    EXIT          No — exit
74    FF 00            LDI    X'00'         Yes — reset to zero
76    A6               PLO    6             -
77    18               INC    8             Count days
78    88               GLO    8             -
79    FF 07            SMI    X'07'         Do days = 7
7B    3A 4F            BNZ    EXIT          No - exit
7D    A8               PLO    8             Reset to Monday
7E    30 4F            BR     EXIT          Exit
             .
             .         TABLE STARTS HERE
             .
80           TABLE     DC     X . . .
```

Fig. 2. The heat-control system software.

ket. But unlike most other chips, it is a CMOS device.

CMOS (complementary metal-oxide semiconductor) logic has several advantages over the older logic families. It uses ultralow power, typically measured in microwatts (uW), and is relatively noise free. It is also very flexible in widely varying voltage-supply situations. The disadvantage of using CMOS is that the individual chips tend to cost more and are slower than their TTL counterparts.

The System

Once the Elf kit was assembled, the next step was a thermostat interface for our heat-control system. The key to interfacing a 5 volt computer with the 240 V ac power that runs through an electric heat thermostat is isolating the two components as much as possible. A thermostat turns heat on and off in response to room temperature. If it can be fooled into thinking that the room is hot, it will turn the heat off. Therefore, a computer-controlled heat source located near the thermostat does the trick.

The simplest source of heat, known to all electronics experimenters, is a resistor. A 10 Ohm, 2 Watt power resistor wired directly to the thermostat's metal plate serves to turn our heat off when power is applied. When the resistor is not powered, the thermostat responds normally to room temperatures, switching on or off as necessary.

So far, the system appears to be fail-safe. If any part of the electronics fails, the thermostats revert to normal operation. However, poor programming of the computer could result in a continuously powered resistor that would prevent the heat from ever switching on. With eight years' programming experience between us, we hope this never happens (of course, we would never admit it even if it did).

The Hardware

Since neither of us knows much about hardware, the circuit developed by trial and error with big assists from Kilobaud Klassroom and Don Lancaster's excellent book *CMOS Cookbook*. There are two parts to the circuit: The first generates interrupts (forcing the COSMAC to branch to its interrupt service routine, which accumulates time); the second part acts as an output device to control the power to the resistors.

To generate interrupts, a 60 Hz signal is taken off the power-supply transformer and fed into a CD4050 buffer with a feedback resistor. This acts as a Schmitt trigger and converts the sinusoidal ac into a square wave. (A Schmitt trigger is a device that turns on at a certain voltage and turns off at a lower voltage. The area between the two voltages is termed *hysteresis*. Schmitt triggers are used primarily to square up sloping waveforms.) The output of the Schmitt trigger is fed into the interrupt flip-flop (½CD4013), and brings the $\overline{\text{INT}}$ line high until the COSMAC acknowledges the interrupt.

The rest of the hardware is devoted to controlling the power resistors. Two CD4076 registers act as an output port. Their output is fed into CD4050 buffers (one for each thermostat), and the output of the buffers is fed, in turn, through the same number of limiting resis-

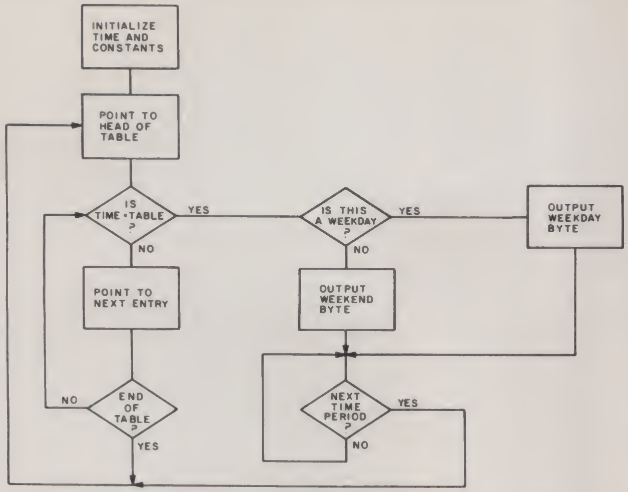


Fig. 3. Flowchart for thermostat control routines.

tors into the base of power transistors, causing the transistors to switch on. One side of each power resistor is connected to a transistor's collector, and the other side is attached to a 5 volt line.

The Elf's power supply was not hefty enough to control more than two or three transistors, so we attached a simple, unregulated 6 volt supply to supplement the resistors.

The Software

The heat-control program requirements were simple: devise a seven-day schedule of events so that given day-of-week and time-of-day information, the computer can decide whether or not to power the thermostat interfaces. Our storage limit was 256 bytes.

In order to decipher day-of-week and time-of-day information, the computer must have its own mechanism for calculating or accumulating the passage of time. We originally planned to use software loops for timing but, after days of experimentation, could not achieve the accuracy needed for real-time control.

To attain that level of accuracy, each side of all branches must have equal execution times, and the computer's timing crystal must be trimmed to a frequency that is a divisor of one second, a job that requires test equipment and a great deal of patience, neither of which we had. A 60-cycle ac line works just as well, if not better.

A simple divide-by-four circuit that brings the line down to 15 Hz is connected to the COSMAC's interrupt line. The interrupt-service routine counts fifteen interrupts for each second, and 360 seconds for each six-minute interval.

Because we chose to accumulate time in a one-byte area to conserve storage, we were limited to six-minute intervals with which to build a daily schedule of events. The 1440 minutes of one day, divided by 256 (or hex FF, the maximum content of one byte), calculates to an interval slightly under six minutes. We rounded to six minutes and numbered the intervals in each day from hex 00 to hex F0 where hex 00 is midnight.

An in-memory table keeps the heat-control schedule. Each entry in the table consists of three bytes. The first byte contains the dispatching time, that is, the time at which a change in the heating is to occur.

The second byte in the table contains the instructions for each thermostat for that particular time. Each bit in this byte corresponds to a different room. If the bit is on, the power resistor for that room will heat up, and the heat will go off. If the bit is off, the heat will go on. The third byte in the table is similar to the first byte except that it is used for the weekend, rather than the weekday.

The program logic is simple. Each entry in the table is compared to the time-of-day information, which is kept in one of COSMAC's 16 general-purpose registers and updated by the interrupt-service routine. If the time of day matches the entry, the day-of-week register, also maintained by the interrupt-service routine, is examined. If it is a weekday, the second byte of the table entry is output. Otherwise, the third byte, the weekend byte, is output. The program continuously scans

Each power resistor is mounted directly on the thermostat. Heat-sink compound was used to assure a good thermal connection.

The power resistor is completely hidden by the plastic cover of the thermostat.

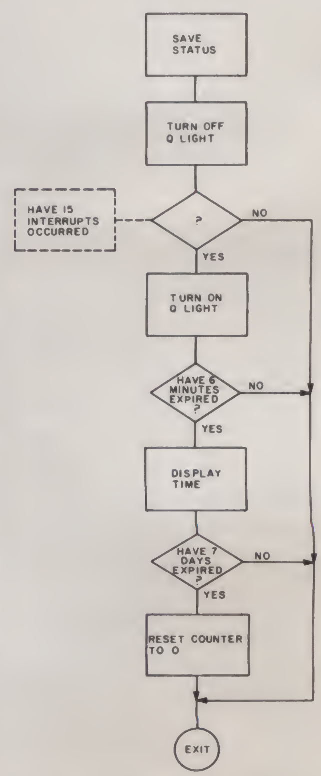


Fig. 4. Flowchart for interrupt routine.

the table in a loop.

Conclusion

The final justification for a computerized heat-control system should be the realization of electric-bill savings. But so far, our satisfaction with our system has been mostly psychological. The Elf heating system is our first home-computing application and we couldn't be happier, although calculating exact savings on our bill has been difficult. Our 1978 heating bill showed a 25 percent drop in kilowatt-hour usage compared to the same period in 1977, but the winter of '77 was colder than that of '78.

The computer turns on the heat an hour before we get out of bed, turns it off while we are away at work, turns it back on an hour before we get home and off again when we go to bed. No longer is it a cold shock to push the blankets aside in the morning and climb out of bed.

When we come home from work in the evening, the kitchen is warm and inviting, and as we cook dinner, our den is being heated. While our various pets and houseplants may shiver a bit during the day when the heat is off and we are away, we are warmer now at a lesser cost.■

The entire system is mounted in the laundry room. The excellent resistance to power-supply problems exhibited by CMOS devices is illustrated by the computer's being plugged into the same outlet as the washing machine.

David Price
3901 Victoria Lane
Midlothian VA 23113

Music, Maestro!

In music, the letters A and D designate notes on the scale; 8 signifies a measure of a note's duration. The AD8 is a computer-controlled synthesizer system from ALF Products.

Imagine this scenario: You have just arrived home from work. Immediately after eating dinner, you run to your computer and load a rather hefty program into it. You then switch on a second peripheral—a computer controlled synthesizer that has been interfaced with your system.

You start the program running. Suddenly, a nearby speaker comes to life, rending the ether with your electronic version of Beethoven's Symphony No. 4.

Ah, yes—the Glorious Fourth. Where there used to be strings and reeds, there are now DACs, oscillators and filters; but the music itself remains the same. The myriad waveforms rise and subside, creating a virtuoso performance.

You have spent weeks working on this piece, writing and rewriting, until each rhythm line, each individual timbre blends perfectly with the others to form a harmonious whole.

You plan to make a tape of it for posterity, and someday perhaps you will cut your first LP. After that, who knows? Maybe you'll go on a concert tour, playing to throngs of devoted fans.

But now, you are simply going to sit back and listen—and enjoy.

General

What *exactly* was that second peripheral? Does it even exist? It does indeed: It is the ALF AD8, a powerful new synthesizer system designed to be controlled directly by a computer.

Most conventional instruments generate sound through mechanical means, such as air passing through a tube or a string being plucked. Synthesizers (such as the AD8) are endowed with incredible versatility because they consist entirely of electronic circuitry, which is more readily controllable than a mechanical device.

Applications

Basically, you can use the AD8 anywhere you would use any other synthesizer. I would like to suggest the following as possible applications.

Composition. Many composers (and would-be composers) do not actually play an instrument; a device such as the AD8 would virtually put an entire orchestra at their disposal. And unlike conventional instruments, which take years to master, an AD8-based computer synthesis system is relatively easy to learn to program.

Exactly how long does it take to orchestrate a piece of music? Philip Tubb, president of ALF, says, "Our tests show that a typical song (after a little practice) can be taken from sheet music and turned into a perfected song in about an hour per minute of real-time playing; and this includes debugging."

Various tools are available with which to ease this process. In particular, current AD8 software (all of which is 8080-based) includes several loaders, drivers and a relocating assembler. Also planned are a high-level language designed to facilitate the creation of computer music and a library of songs.

Another area of interest is computerized composition. A great deal of work has already been done in this field (the famed Illiac IV pieces are the first that come to mind), but I do not propose to show how these efforts might be applied to microcomputers. Nevertheless, it is an interesting thought.

Live performance. One particular difficulty with onstage synthesizers is the problem of changing the sound parameters. Patch cords, matrix switches, punched cards and similar controls take time to change. By connecting an organ keyboard and several

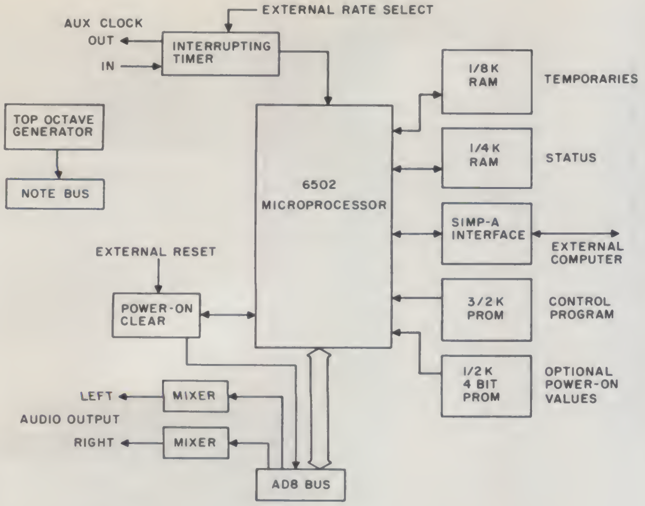


Fig. 1. AD8 control board block diagram.

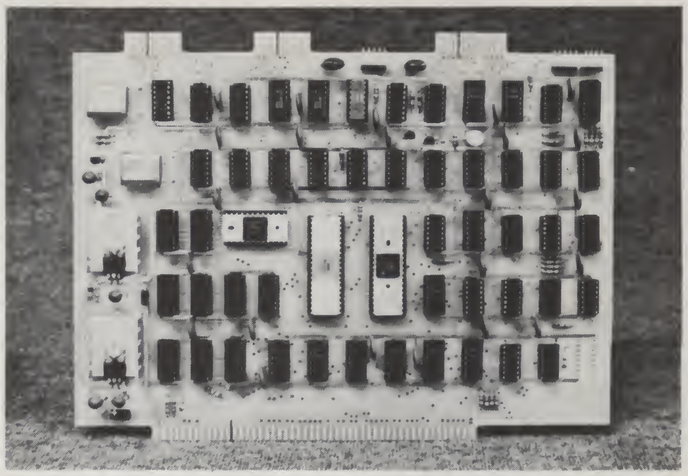


Photo 1. Control board. Large ICs are the 6502 microprocessor and 6820 I/O port.

controls to your external computer system, you can use the normal playing method. The controls can still be used to alter sound parameters as on a regular synthesizer. For significant changes in the sound characteristics, where normally all parameters would have to be changed with time-consuming adjustments, the computer can change all parameters to preprogrammed settings. Any or all adjustments can be changed instantly.

For example, one rotary control can be used to select the different sounds needed throughout a performance. By simply changing that one control, the computer can change as many parameters as necessary to shift from the previous sound to the new one. This can even be done between notes in a song. You can also make the simple changes manually as you go, just as you would on any synthesizer.

The AD8 with all eight synthesis boards is essentially eight separate synthesizers, each with its own set of sound parameters, including waveform and envelope. A couple of channels (synthesis boards, each of which is monophonic) could do a preprogrammed background (either a single repeating note sequence or a whole melody with no repeats, entered in advance by playing it on the keyboard) while you use the other channels for live performance with the keyboard. Or, have the computer playing the notes while you adjust the sound parameters live. You can have more than one channel playing the notes you press, with each channel having a different waveform (to sound like different instruments) or some delayed for an echo or reverb effect. The speed of the computer allows endless variations of control.

Studio recording. Although synthesizers have found great success in live performance, they are infinitely more comfortable within the realm of the recording studio. This is because conventional synthesizers are monophonic devices, and hence can only produce polyphonic music through sound-on-sound and multitrack recording techniques.

The AD8 overcomes this limitation because its octophonic capabilities allow the piece to be recorded in stereo immediately after being programmed. And the two-waveform-memories-per-board feature makes it possible to simulate an infinite number of tracks, as long as no more than eight are played simultaneously.

Because the AD8 is based on a modular, computer-like bus, the hobbyist user can start with a minimum configuration and expand incrementally from there until he finally has an instrument comparable to one of those mammoth "studio" synthesizers.

The basic system consists of four components: a controller board, a power supply, a backplane board and one or more synthesis boards.

Controller Board

The AD8 controller board (see Fig. 1 and Photo 1) handles all communications with the host computer. It uses an onboard MOS Technology 6502 microprocessor with 128 bytes of RAM and 1536 bytes (1.5K) of PROM.

Two Molex KK-100 connectors are utilized. One of them acommodates the front-panel functions; the other connects to the phono jacks on the rear panel, providing right- and left-channel stereo outputs.

Provisions are made for an orderly power-up sequence. Upon power-up (or whenever the Reset switch is depressed), a signal is sent to all boards via bus line $\overline{\text{POC}}$ (Power-On Clear). Then the microprocessor sends initialization parameters to all devices. The reset function can also be activated from the host computer. Along with RESET, the computer can also issue the following commands to the

ALF Products, Inc. (AD8 synthesizers)
128 South Taft
Lakewood CO 80228
PAIA Electronics, Inc. (Low-cost synthesizer kits)
1020 West Wilshire Blvd.
Oklahoma City OK 73116
Aries, Inc. (Aries System 300 synthesizer)
119 Foster St.
Peabody MA 01960
Electronic Music Laboratories, Inc. (Electrocomp synthesizers)
PO Box H
Vernon CT 06066
ARP Instruments, Inc. (ARP synthesizers)
320 Needham St.
Newton MA 02164
Moog Music, Inc. (Moog synthesizers)
PO Box 131
Williamsville NY 14221
Computer Music Journal
c/o People's Computer Company
Box E
Menlo Park CA 94025
Electronotes Magazine
203 Snyder Hill Road
Ithaca NY 01960

Table 1. Addresses worth having.

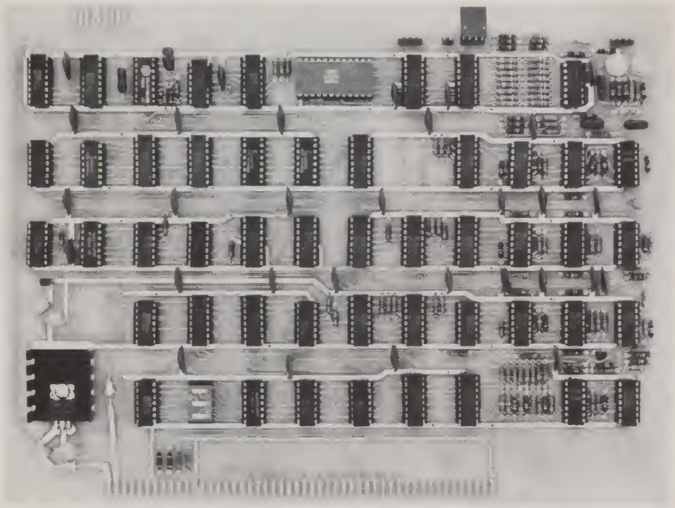


Photo 2. Synthesis Board. Portion to the right of bottom edge connector is the analog section (note the many resistors and other discrete components in this section), which is separated from the digital section (to the left) by a ground trace on the circuit side of the board (not shown). Top octave note bus is at top left; AD8 Bus at bottom. Small Molex connectors at top are used for options and connection to accessories or external synthesizers.

controller:

MASTER CLOCK. Sets the number of clock cycles to count before issuing an interrupt to the computer.

SWITCHES/STEP. Allows the front-panel switches to be read. Also allows a specified number of steps to be executed (similar to a "single-step" mode).

MEMORY. Allows a byte of the 6502's memory to be read or written.

DISPLAY. Allows a function being monitored on any of the three front-panel displays to be modified.

TIMING. Used with the SEQUENCE command.

SEQUENCE. Indicates the memory address (for the 6502) where a set of commands is stored. These commands will be executed after the number of clock cycles specified by the TIMING command.

JUMP. Causes the 6502 to jump to the specified address.

ADDRESS. Specifies the memory address to be accessed by the MEMORY command.

MOVE. Copies one waveform memory onto another.

MODE. Allows LOW-PASS FILTER and/or CYCLE commands to be generated automatically upon receipt of a PITCH command.

In addition to processing commands from the computer, the controller also scans accessories. Commands can be issued by any accessory connected to the AD8 bus, and they will be treated as if they were sent by the computer itself.

Synthesis Board

The AD8 synthesis board (see Fig. 2 and Photo 2) has a range of 96 notes (eight octaves). It allows complete software control over pitch, volume, attack, decay, sustain, waveform, stereo location and low-pass filtering.

The AD8 can handle up to eight synthesis boards. Since each board is capable of producing one tone at a time, an eight-board system is actually an *octophonic* synthesizer. (Photo 3 shows a complete eight-board system.)

Each synthesis board contains two waveform memories. Each memory contains 64 bytes that define one cycle of the waveform. The user defines the wave by specifying the amplitude of the wave (on a scale from 0 to 255) at 64 evenly spaced points.

Why *two* waveform memories? Well, suppose you had a square wave stored and you wanted to switch to a sine wave in midstream (as in Fig. 2a). With two memories, you could do this by programming them with a square wave and a sine wave, respectively, and then changing the WAVEFORM SELECT value (see below). If you had only one memory, you would have to reprogram the waveform *as it was being played* (see Fig. 2b).

Using the controller-board

Fig. 2a.

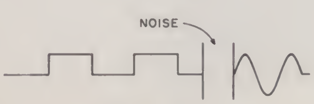


Fig. 2b.

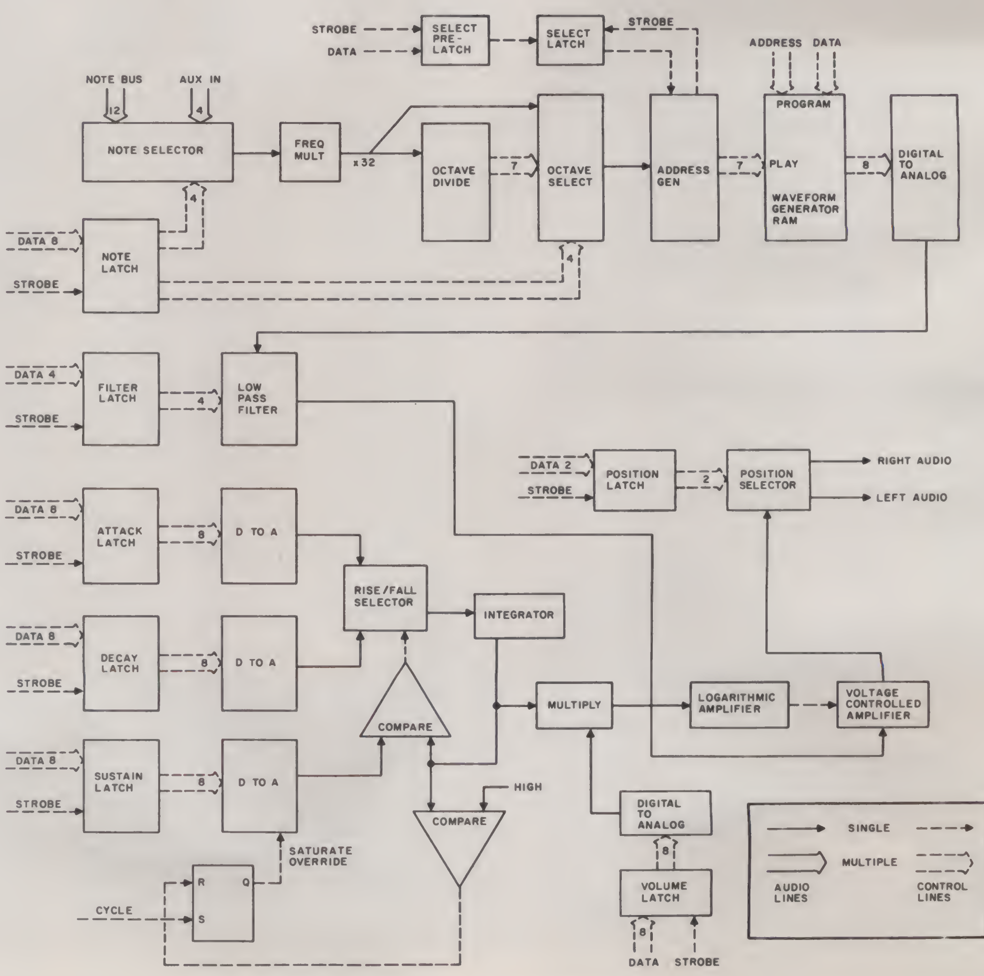


Fig. 2. AD8 synthesis board block diagram.

frequency outputs, the synthesis board has a frequency range from about 27 Hz to 6600 Hz. This is equivalent to the pitch range of a standard piano. There are 96 notes, all equally tempered and tuned to A = 440. The AD8 provides an additional eight notes beyond the highest of a piano, however. By defining more than one cycle in memory, even higher frequencies can be produced (albeit with decreased waveform accuracy).

The following synthesis board commands are available.

VOLUME. Selected on a scale from 0 to 255.

LOCATION. Two bits of stereo location control allow the tone to be output on either, neither or both stereo channels.

PITCH. One bit selects "rest" or "play" mode. Three bits select the octave, and four bits select one of 16 frequency inputs. The controller board provides only 12 frequencies, since there are normally only 12 notes per octave. The remaining four can be supplied by the user, if desired.

ATTACK, DECAY, SUSTAIN and CYCLE (see Fig. 3). Upon receipt of the CYCLE command, the loudness of the tone is set to zero and raised to the VOLUME level at the rate specified by the ATTACK value. Then, it is lowered to the SUSTAIN level at the rate specified by the DECAY value. An additional factor in defining the envelope is the "release" value. This is the rate at which the loudness is decreased from the sustain level back to zero. If a release value is desired, another phase must be added: After completion of the normal ADS cycle, the decay rate is reset to the desired release rate, and the sustain value is reset to zero. 256 different attack rates are available, ranging from 4 ms to 1.3 seconds. The sustain level is specified as a percentage of the volume level on a scale from 0 to 255. There are also 256 different decay values, ranging from 3 ms to 7.8 seconds.

LOW-PASS FILTER. The low-pass filter reduces the loudness of frequencies higher than a specified level (see Fig. 4). The user may select from 16 cutoff frequencies. The MODE controller command allows a new low-pass-filter value to be set automatically as the pitch is changed. The filter used in the AD8 is a two-pole Butterworth.

WAVEFORM SELECT. Selects which of the two memories will be "active" (the one used to produce the tone).

The audio output of the synthesis board is sent to a Molex KK-100 connector at the top of the board, where it is normally jumpered to another line on the same connector. From there, it runs through the stereo location circuitry, and is then routed to the phono jacks on the rear panel.

The jumper may be removed from the connector, allowing the user to intercept the signal and send it to an accessory device, such as an external (Moog-type) synthesizer.

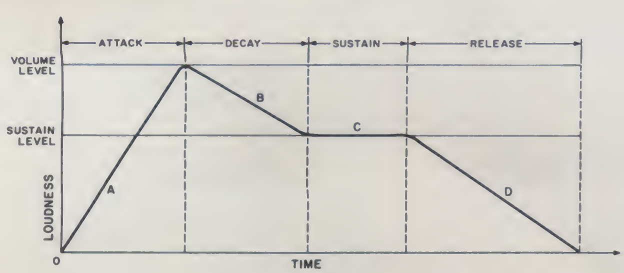


Fig. 3. The ABCDs of ADSR. (A) At the beginning of the note, the loudness of the tone rises to the volume level at the rate specified by the attack value. (B) The loudness then falls to the sustain level at the rate specified by the decay value. (C) Loudness remains at the sustain level. (D) The sustain level is reset to zero, and the decay rate is reset to the desired release rate. Loudness falls from the old sustain level to zero at the rate specified by the new decay (i.e., release) value.

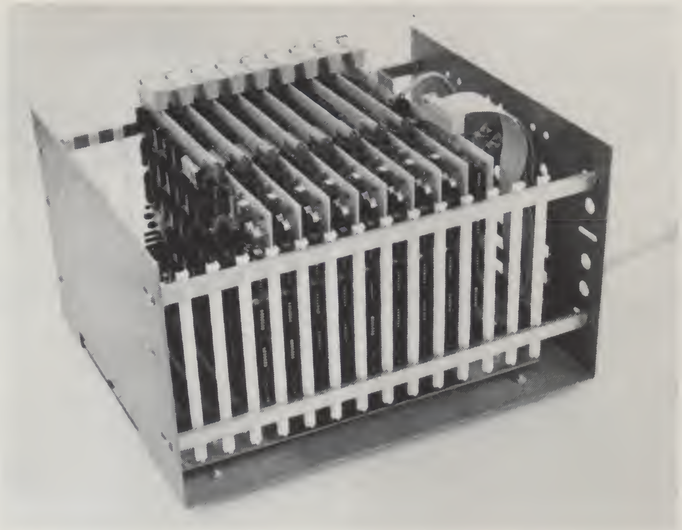


Photo 3. AD8 full system with control board, eight synthesis boards, and backplane board. (Shown in prototype S-100 card frame.)

Interface

The AD8 and the host computer communicate via a special 10-bit interface, christened SIMP (Standard Interface for Micro-Peripherals). Eight of the ten lines are used to transmit the byte-sized data word; the remaining two bits indicate which "channel" is being utilized.

There are four channels: the function channel, the waveform channel, the request channel and the data channel. When a byte is transmitted to the function, waveform or request channel, the controller stores that value as the Current Process ID.

AD8 commands are encoded in the form XXY (octal), where XX is the command code, and Y is the synthesis board whose parameters are being accessed. For example, to reset the volume level of board #2, the computer would first send a 102_8 to the function channel. This would indicate volume (10Y) and board #2 (XX2). The value 102_8 would then become the Current Process ID.

When a byte is transmitted to the data channel, that byte is treated as a parameter for the Current Process ID (in this case, the data byte would indicate the new volume level for board #2).

The waveform channel is used to indicate the waveform memory element to be sent next over the data channel. An auto-increment feature allows consecutive elements to be transmitted after specifying only the first address.

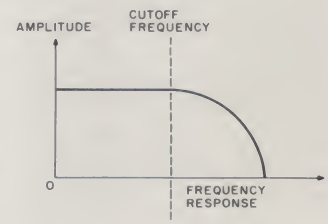


Fig. 4. A low-pass filter passes any frequency lower than the cutoff frequency. Frequencies higher than the cutoff frequency are reduced proportionally in amplitude.

The request channel allows the computer to interrogate the AD8. For example, sending a 102_8 over the request channel would cause the AD8 to return the current volume value of board #2 over the data channel.

An S-100-compatible SIMP interface is available from ALF. If yours is not an S-100 mainframe, a home-brew SIMP interface can be constructed from a pair of M6820s (details available from ALF). The 25 SIMP lines are defined as follows:

ID0-ID7: Eight lines from the AD8 to the computer. ID0 is the least significant bit (LSB).

IC0-IC1: Two lines that define

the channel number of the input byte. IC0 is the LSB.

$\overline{IDA}$: Pulled low by the AD8 when input data is present, and returned to its normal (high) state after a negative transition of IAC.

IAC: A negative transition is created by the computer to indicate that the input data has been received.

The following lines are identical to their counterparts above, except that data flows from the computer to the AD8:

OD0-OD7
OC0-OC1
$\overline{ODA}$
OAC

The twenty-fifth line is GROUND.

Lines $\overline{IDA}$, IAC, $\overline{ODA}$ and OAC above are used to provide ready/busy signals. $\overline{IDA}$ indicates that input is present. IAC indicates that the input has been received (by the computer). $\overline{ODA}$ indicates that output is present. OAC indicates that the output has been received (by the AD8).

SIMP connects to the AD8 controller board via the DB-25P connector on the rear panel. The *maximum* recommended cable length is three meters. Since the AD8 unit is generally positioned directly beside the host computer, however, this shouldn't be of any consequence to most users.

Configuration

The AD8 backplane board is designed to accommodate up to 14 individual boards. It is similar to the S-100 backplane in that it uses two rows of 50-conductor connectors, with .125 inch spacing, for a total of 100 lines.

At the front of the board, connections are provided for the front panel. Provisions are made on the rear for running cables to an additional backplane. Extender backplanes provide only 13 slots, however, because one slot is occupied by a driver board.

Each backplane has its own power supply. Power connections to the backplane board are through Molex KK-156 connectors. A six-conductor connector allows a maximum of 28 Amps at +8 V, and four Amps at +16 V and −16 V. A seven-conductor connector provides six conductors for ground and one conductor for a 120 Hz clock input from the secondary of the +8 V transformer.

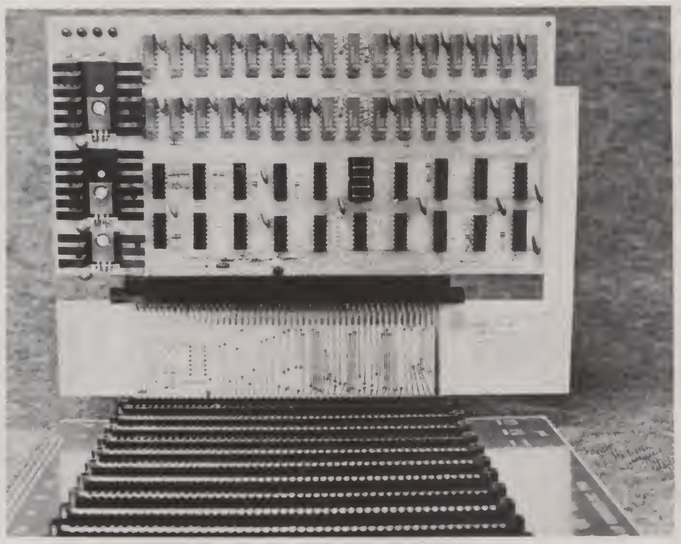


Photo 4. S-100 memory board. Shown in 5-1-1 backplane board. Allows standard S-100 memories (with no wait states) to be used for waveform and routine storage. Shown with Imsai 4K memory board.

It is possible to use S-100 memory boards with the AD8 bus by plugging a special adapter into the backplane (Photo 4) and then plugging the board itself into the adapter. (Warning: The memories must have *no* wait states.)

Two front panel options, a blank front panel or a full front panel, are available. The blank front panel contains only two switches: Power and Reset. The full front panel includes both of these, plus the following:

Run/Step. Allows the music to be run through a predetermined number of steps at a time.

Rate. Allows the rate to be varied from as slow as one-fourth to as fast as quadruple the normal rate.

Mixing. An eight-channel mixer. Allows the volume of each channel to be individually adjusted.

Display. Selects the parameters to be monitored by the three status displays.

Disable. Allows the Reset, Rate, or Run/Step switches to be disabled.

The full front panel also provides the following indicators:

Power-On Clear. Also indicates a Reset operation.

Disable. Indicates that the Reset, Rate, and Run/Step switches have been disabled.

Output. Monitors the output of the eight synthesizer channels.

Status Displays (3). Each has eight data LEDs and one LED that flashes when the data is updated.

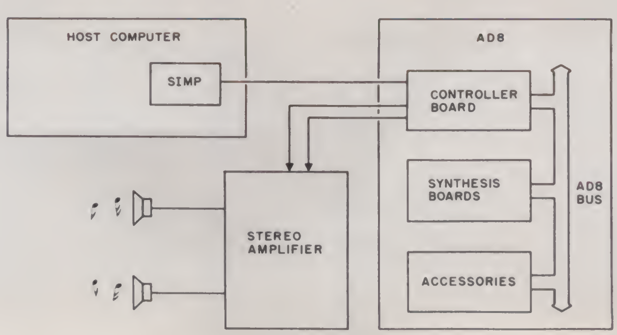


Fig. 5. Standard system configuration.

The AD8 does not contain its own amplifier. Instead, the audio outputs from the phono jacks should be connected to the user's stereo system.

Hobbyist System

Up to this point, I have been describing the "professional" version of the AD8 (shown in Fig. 5 and Photo 3). However, there is also a substantially less expensive hobbyist-oriented AD8 system. *No quality compromises are made in the hobbyist version.* There are no differences in terms of performance, either; in fact, the synthesis board is identical in both systems (with not-worth-mentioning minor differences).

The hobbyist system differs from the professional system in the following respects:

1. SIMP is not used. Instead, the controller board is inserted directly into a slot of the S-100 bus. It is addressed as a 2K memory, and controlled by the computer through this memory. Any 2K boundary may be used (switch-selectable).

2. It does not have its own power supply. Instead, it uses the computer's.

3. It does not use a backplane; the boards are connected by ribbon cables.

4. It is not compatible with the standard AD8 bus. It uses, instead, two separate buses: the AD8 Micro-Bus and the Note bus.

5. If the host computer uses a 2 MHz clock and brings this signal out through line 49 ($\overline{CLOCK}$), then the top octave generator can be driven directly by it. Otherwise, the user must purchase the crystal oscillator option.

6. Accessory devices cannot issue commands.

7. Although functionally identical, the synthesis board is slightly different internally from that of the professional system.

Coda

Now that you know that second peripheral exists, the computer-composition scenario could become a reality. ■

TRS-80 Owners~ Read this ad and save over $100!

Special Introductory Price: **$395**

Introducing the Vista V80 Mini Disk System

- ☐ **23% MORE STORAGE CAPACITY** — Increases your usable storage capacity 23% from 55,000 to 67,800 bytes on drive one.
- ☐ **FASTER DRIVE** — Electronically equal to the TRS-80 Mini-Disk System, but up to 8 times faster (Track-to-track access in 5ms for the V80 versus 40ms for TRS-80).
- ☐ **DOES NOT VOID TRS-80 WARRANTY** — V80 also has 90-day warranty.
- ☐ **HERE'S WHAT YOU GET:** Minifloppy disk drive/Power Supply/Regulator board/Compact case
- ☐ **DOUBLE DENSITY FOR DOUBLE STORAGE** — The V80 will work with the Vista double-density expansion unit when available.
- ☐ **SHIPPED TO YOU READY TO RUN** — Simply take it out of the box, plug it in and you're ready to run.

MONEY-SAVING COUPON

To: Vista Computer Company, Dept. K29
2909 Oregon Court, Torrance, CA 90503
(213) 320-3880

Vista V20

Please send me ________ V80 unit(s). @ $395.
This is my: ☐ 1st ☐ 2nd ☐ 3rd ☐ 4th disk drive. (If this is your 1st drive, you need a connecting cable at $29.95.)
Check here for cable: ☐
(In California, add 6% sales tax.)

I've enclosed: ☐ Check ☐ Money Order
Please charge to: ☐ Master Charge ☐ Visa Card No.________________
Card Expiration Date________________ Signature________________

Name ________________________________
Address ________________________________
City ________________ State/Zip ________________

Reader Service—see page 179

Adrian R. Thornton
6404 Overton Rd.
Louisville KY 40228

Madam Dupré's House of the Zodiac

"What's your sign?" Are you tired of hearing that inane question posed everywhere you go? Try springing this "horoscope" on the next person who asks.

```
RUN

WELCOME TO MADAM DUPRE'S  'HOUSE OF THE ZODIAC'
TELL ME YOUR BIRTHDAY AND I WILL PRESENT YOUR HOROSCOPE

ENTER YOUR BIRTH MONTH AND DAY AS 4 DIGITS (MMDD): 0912

                Y O U R   S I G N   I S
                      V I R G O
                      - - - - -

YOU ARE THE LOGICAL TYPE AND HATE DISORDER.  THIS NITPICKING IS
SICKENING TO YOUR FRIENDS.  YOU ARE COLD AND UNEMOTIONAL AND
SOMETIMES FALL ASLEEP WHILE MAKING LOVE.  VIRGOS MAKE GOOD BUS
DRIVERS.
END AT LINE  8040
*

 RUN

WELCOME TO MADAM DUPRE'S  'HOUSE OF THE ZODIAC'
TELL ME YOUR BIRTHDAY AND I WILL PRESENT YOUR HOROSCOPE

ENTER YOUR BIRTH MONTH AND DAY AS 4 DIGITS (MMDD): 1228

                Y O U R   S I G N   I S
                   C A P R I C O R N
                   - - - - - - - - -

YOU ARE CONSERVATIVE AND AFRAID OF TAKING RISKS.  YOU DON'T
DO MUCH OF ANYTHING AND ARE LAZY.  THERE HAS NEVER BEEN A
CAPRICORN OF ANY IMPORTANCE.  CAPRICORNS SHOULD AVOID STANDING
STILL TOO LONG AS THEY TEND TO TAKE ROOT AND BECOME TREES.
END AT LINE  12040
*
```

Sample runs.

Every third person you meet seems to want to know your "sign" and, whether requested to or not, proceeds to tell you what kind of person you are and why you both are or are not compatible.

Having the following program you can just smile, lead him/her to your terminal and let Madam Dupré determine your friend's horoscope. Based on the user's birth date, the program generates a humorous, sarcastic and surprisingly accurate horoscope appropriate to the user's "sign."

The program itself is simple and concise. The idea is based on a clever spoof of horoscopes whose author(s) I have been unable to determine but gladly give credit to for inspiring this program.

The program was written on a Heathkit H8 computer with 16K of memory using Heath's Extended Benton Harbor Cassette BASIC.■

```
00010 REM ****************************** HOROSCOPE ********************************
00020 REM
00030 REM  ADAPTED TO BASIC BY ADRIAN R. THORNTON -- AUG., 1978
00040 REM
00050 DATA "A Q U A R I U S","- - - - - - - -","P I S C E S","- - - - - -"
00060 DATA "A I R E S","- - - - -","T A U R U S","- - - - - -"
00070 DATA "G E M I N I","- - - - - -","C A N C E R","- - - - - -"
00080 DATA "L E O","- - -","V I R G O","- - - - -"
00090 DATA "L I B R A","- - - - -","S C O R P I O","- - - - - - -"
00100 DATA "S A G I T T A R I U S","- - - - - - - - - - -"
00110 DATA "C A P R I C O R N","- - - - - - - - -"
00120 REM
00130 PRINT :PRINT "WELCOME TO MADAM DUPRE'S  'HOUSE OF THE ZODIAC'"
00140 PRINT "TELL ME YOUR BIRTHDAY AND I WILL PRESENT YOUR HOROSCOPE"
00150 PRINT :INPUT "ENTER YOUR BIRTH MONTH AND DAY AS 4 DIGITS (MMDD): ";B
00160 REM
00170 IF B<0101 OR B>1231 THEN PRINT "INVALID DATE ENTRY, TRY AGAIN.":GOTO 150
00180 IF B<0120 THEN S=12:GOSUB 340:GOTO 12000
00190 IF B<0219 THEN S=1:GOSUB 340:GOTO 1000
00200 IF B<0321 THEN S=2:GOSUB 340:GOTO 2000
00210 IF B<0420 THEN S=3:GOSUB 340:GOTO 3000
00220 IF B<0521 THEN S=4:GOSUB 340:GOTO 4000
00230 IF B<0621 THEN S=5:GOSUB 340:GOTO 5000
00240 IF B<0723 THEN S=6:GOSUB 340:GOTO 6000
00250 IF B<0823 THEN S=7:GOSUB 340:GOTO 7000
00260 IF B<0923 THEN S=8:GOSUB 340:GOTO 8000
00270 IF B<1023 THEN S=9:GOSUB 340:GOTO 9000
00280 IF B<1122 THEN S=10:GOSUB 340:GOTO 10000
00290 IF B<1222 THEN S=11:GOSUB 340:GOTO 11000
00300 S=12:GOSUB 340:GOTO 12000
00310 REM
00320 REM PRINT PLAYER'S SIGN SUBROUTINE
00330 REM
00340 PRINT :PRINT
00350 FOR I=1 TO S
00360 READ S1$:READ S2$
00370 NEXT I
00380 PRINT TAB(19);"Y O U R   S I G N   I S"
00390 PRINT TAB((80-LEN(S1$))/2-9);S1$
00400 PRINT TAB((80-LEN(S2$))/2-9);S2$
00410 PRINT
00420 RETURN
00970 REM
00980 REM AQUARIUS SUBROUTINE
00990 REM
01000 PRINT "YOU HAVE AN INVENTIVE MIND AND ARE INCLINED TO BE PROGRESSIVE."
01010 PRINT "YOU LIE A GREAT DEAL.  ON THE OTHER HAND YOU ARE INCLINED TO BE"
01020 PRINT "CARELESS AND IMPRACTICAL CAUSING YOU TO MAKE THE SAME MISTAKES"
01030 PRINT "OVER AND OVER AGAIN.  PEOPLE THINK YOU ARE STUPID."
01040 END
01070 REM
01080 REM PISCES SUBROUTINE
01090 REM
02000 PRINT "YOU HAVE A VIVID IMAGINATION AND OFTEN THINK YOU ARE BEING"
02010 PRINT "FOLLOWED BY THE 'CIA' OR 'FBI'.  YOU HAVE MINOR INFLUENCE OVER"
02020 PRINT "YOUR ASSOCIATES AND PEOPLE RESENT YOU FOR YOUR FLAUNTING OF YOUR"
02030 PRINT "POWER.  YOU LACK CONFIDENCE AND ARE GENERALLY A COWARD."
02040 PRINT "PISCES PEOPLE DO TERRIBLE THINGS TO SMALL ANIMALS."
02050 END
02970 REM
02980 REM AIRES SUBROUTINE
02990 REM
03000 PRINT "YOU ARE THE PIONEER TYPE AND HOLD MOST PEOPLE IN CONTEMPT."
03010 PRINT "YOU ARE QUICK TEMPERED, IMPATIENT AND SCORNFULL OF ADVICE."
03020 PRINT "YOU ARE NOT VERY NICE."
03030 PRINT
03040 END
03970 REM
03980 REM TAURUS SUBROUTINE
03990 REM
04000 PRINT "YOU ARE PRACTICAL AND PERSISTENT.  YOU HAVE A DOGGED DETERMINATION"
04010 PRINT "AND WORK LIKE HELL.  MOST PEOPLE THINK YOU ARE STUBBORN AND BULL"
04020 PRINT "HEADED.  YOU ARE A COMMUNIST."
04030 PRINT
04040 END
04970 REM
04980 REM GEMINI SUBROUTINE
04990 REM
05000 PRINT "YOU ARE A QUICK AND INTELLIGENT THINKER.  PEOPLE LIKE YOU"
05010 PRINT "BECAUSE YOU ARE BISEXUAL.  HOWEVER, YOU ARE INCLINED TO EXPECT"
05020 PRINT "TOO MUCH FOR TOO LITTLE.  THIS MEANS YOU ARE CHEAP.  GEMINIS ARE"
05030 PRINT "KNOWN FOR COMMITTING INCEST."
05040 END
05970 REM
05980 REM CANCER SUBROUTINE
05990 REM
06000 PRINT "YOU ARE SYMPATHETIC AND UNDERSTANDING OF OTHER PEOPLE'S PROBLEMS."
06010 PRINT "THEY THINK YOU ARE A SUCKER.  YOU ARE ALWAYS PUTTING THINGS OFF."
06020 PRINT "THAT'S WHY YOU'LL NEVER MAKE ANYTHING OF YOURSELF.  MOST WELFARE"
06030 PRINT "RECIPIENTS ARE CANCER PEOPLE."
06040 END
06970 REM
06980 REM LEO SUBROUTINE
06990 REM
07000 PRINT "YOU CONSIDER YOURSELF A BORN LEADER.  OTHERS THINK YOU ARE PUSHY."
07010 PRINT "MOST LEO PEOPLE ARE BULLIES.  YOU ARE VAIN AND DISLIKE HONEST"
07020 PRINT "CRITICISM.  YOUR ARROGANCE IS DISGUSTING.  LEO PEOPLE ARE KNOWN"
07030 PRINT "THIEVES."
07040 END
07970 REM
07980 REM VIRGO SUBROUTINE
07990 REM
08000 PRINT "YOU ARE THE LOGICAL TYPE AND HATE DISORDER.  THIS NITPICKING IS"
08010 PRINT "SICKENING TO YOUR FRIENDS.  YOU ARE COLD AND UNEMOTIONAL AND"
08020 PRINT "SOMETIMES FALL ASLEEP WHILE MAKING LOVE.  VIRGOS MAKE GOOD BUS"
08030 PRINT "DRIVERS."
08040 END
08970 REM
08980 REM LIBRA SUBROUTINE
08990 REM
09000 PRINT "YOU ARE THE ARTISTIC TYPE AND HAVE A DIFFICULT TIME WITH REALITY."
09010 PRINT "IF YOU ARE A MAN YOU MORE THAN LIKELY ARE QUEER.  CHANCES FOR"
09020 PRINT "EMPLOYMENT AND MONETARY GAINS ARE EXCELLENT.  MOST LIBRA WOMEN"
09030 PRINT "ARE GOOD PROSTITUTES.  ALL LIBRAS DIE OF VENERIAL DISEASE."
09040 END
09970 REM
09980 REM SCORPIO SUBROUTINE
09990 REM
10000 PRINT "YOU'RE SHREWD IN BUSINESS AND CONNOT BE TRUSTED.  YOU SHALL"
10010 PRINT "ACHIEVE THE PINNACLE OF SUCCESS BECAUSE OF YOUR TOTAL LACK OF"
10020 PRINT "ETHICS.  MOST SCORPIO PEOPLE ARE MURDERED."
10030 PRINT
10040 END
10970 REM
10980 REM SAGITTARIUS SUBROUTINE
10990 REM
11000 PRINT "YOU ARE OPTOMISTIC AND ENTHUSIASTIC.  YOU HAVE A RECKLESS"
11010 PRINT "TENDENCY TO RELY ON LUCK SINCE YOU LACK TALENT.  THE MAJORITY"
11020 PRINT "OF SAGITTARIANS ARE DRUNKS OR DOPE FIENDS.  PEOPLE LAUGH AT"
11030 PRINT "YOU A GREAT DEAL."
11040 END
11970 REM
11980 REM CAPRICORN SUBROUTINE
11990 REM
12000 PRINT "YOU ARE CONSERVATIVE AND AFRAID OF TAKING RISKS.  YOU DON'T"
12010 PRINT "DO MUCH OF ANYTHING AND ARE LAZY.  THERE HAS NEVER BEEN A"
12020 PRINT "CAPRICORN OF ANY IMPORTANCE.  CAPRICORNS SHOULD AVOID STANDING"
12030 PRINT "STILL TOO LONG AS THEY TEND TO TAKE ROOT AND BECOME TREES."
12040 END
```

Program listing.

It's There—But Where?

This article simplifies data storage and retrieval techniques, and describes some tricks.

S. J. Mathis, Jr.
1363 Birch Hill Rd.
Mountainside NJ 07092

The biggest difference between programming a micro and programming a larger computer is normally the amount of memory you have at your disposal. This means you have to try to find ways to be reasonably efficient when you're using memory space.

This article describes some of the tricks you can use to store fairly complex information in a limited space. The techniques are general and apply to any microcomputer. To keep things simple we will generally stick with BASIC, assuming most people will use that or can easily translate it to their own machine language. Starting with arrays, we'll work our way through some variations, then have a look at ragged tables (defined after arrays) and, finally, try some simple list structures.

Arrays

An array is simply a table where information is kept. It may contain one or more dimensions but doesn't present any special problems until it has more than one dimension. It's not much trouble to store information in a single-dimensional array; just set aside some space in memory and keep track of the starting point of the array. For example, if we wanted to store 16 items and later access the fourth one, in BASIC we would establish an array (DIM S(16)) and retrieve the fourth element (S(4)), since the language itself keeps track of where things are.

Even if your BASIC doesn't have arrays, you can use a special function to access memory space, as described in Tom Pittman's article ("Tiny BASIC," p. 34) in the Jan. 1977 issue of *Kilobaud*. In terms of absolute memory locations, we might reserve 16 spaces, 0001 through 0016 (in decimal), for example, and, after storing our data, we would get the fourth item back by accessing 0001 + 3.

It's a little easier to store and retrieve things by referencing 0000 + I, where I is the number of the item we want, or, for the fourth item, 0000 + 4. Remember that 0000 is *not* part of the array. In most BASICs, arrays start with S(1), not S(0), so we'll stick with S(1) as the first entry for our examples.

Two-dimensional arrays are a little more interesting. Now we have to keep track of both rows of columns. In BASIC, all we do is DIM A(4,4) and refer to A(2,3) to get to a particular spot in the array.

In machine language or assembly code, we reserve the same 16 spaces, but we need a formula to find our way to an entry in the array. The formula depends on how we want to store the array—by rows or by columns. Suppose we store it by rows. Then our array, shown in Fig. 1a, will look like Fig. 1b in memory. You can see that the first index refers to the row and the second to the column. If we stored it by columns, it would look like Fig. 1c in memory.

Some of the entries are in the same location, but most are placed differently. The one for (2,3) is in the seventh spot when we store by rows and in the tenth spot when we store by columns. The storage-location formula for an entry is not difficult. If we are storing by rows, we need to know how many columns are in the array. If there are C columns, then the location of the (I,J) entry in the array will be found in storage location C*(I − 1) + J; in BASIC you would use

```
100 LET L=(I-1)*C+J
```

and then refer to the 0000 + L position in the array.

By analogy, you should be able to work out the formula for the case where the array is stored by columns. Try it. Why do you need both? Well, you probably don't, but sometimes you'll be working with data in rows and sometimes in columns, and when you're loading the data into your array, you don't want to stop to compute an entry point for each point for each piece of data. You simply want to dump it in, row by row or column by column, however you happen to get it. Usually you'll find *that one way is better* for the particular job at hand, and that you will use both ways.

We occasionally need arrays of more than two dimensions. One place that they are handy is in sorting or categorizing data. If you wanted to divide a group of people into short and tall, fat and thin, male and female, and blond and brunette . . . all at the same time, you could set up a 2 × 2 × 2 × 2 array to hold your counts and then answer questions as to how many short, thin, female blondes were in the group.

The formula for finding the entry location for such an array is an extension of the two-dimensional case. Suppose we have an array with dimensions P, Q, R and S, stored *row-wise*, and we want to find location A(I,J,K,L) in storage. Row-wise in higher-dimensional arrays means that the rightmost index varies *fastest* or first—or, in other words, you cycle all the way through the index on the right before you increment the index on the left of it.

A 3 × 3 × 3 row-wise array is stored as (1,1,1), (1,1,2), (1,1,3), (1,2,1), (1,2,2) . . . and so on up to (3,3,2), (3,3,3). To extend the

1,1	1,2	1,3	1,4
2,1	2,2	2,3	2,4
3,1	3,2	3,3	3,4
4,1	4,2	4,3	4,4

Fig. 1a. An array on paper.

1,1	1,2	1,3	1,4	2,1	2,2	2,3	2,4	3,1	3,2	3,3	3,4	4,1	4,2	4,3	4,4

Fig. 1b. An array stored by rows.

1,1	2,1	3,1	4,1	1,2	2,2	3,2	4,2	1,3	2,3	3,3	4,3	1,4	2,4	3,4	4,4

Fig. 1c. An array stored by columns.

formula, you subtract one from the last index on the right, multiply the whole thing by the new dimension size and add the new index variable to the result. For example, for an array of two dimensions (P,Q), we would use

```
100 LET N=(I-1)*Q+J
```

For one of three dimensions (P,Q,R), we would use the above, followed by

```
110 LET N=(N-1)*R+K
```

And for our four-dimensional array, we use both lines, followed by

```
120 LET N=(N-1)*S+L
```

It's not hard to see why the first one works; the (I – 1)•Q term moves you past the rows you don't want, then the J term gets you over to the right column. If you simplify the second formula by substituting the first line into the second and then multiplying out the terms, you should be able to see what is going on there, too. If the first index is *planes*, the second *rows* and the third *columns*, then the (I – 1)•Q•R term will get you to the right plane, the (J – 1)•R term will skip the unwanted rows and the K will pick the right column for you.

```
100 FOR I=1 TO 4
110 FOR J=1 TO 5
120 LET A(I,J)=0
130 NEXT J
140 NEXT I
```

Fig. 2a. Initializing a two-dimensional array.

```
100 FOR L=1 TO 20
110 LET A(L)=0
120 NEXT L
```

Fig. 2b. Another way to initialize the array.

The last set of lines do the same thing, except it's a little hard to describe geometrically. You could say that the first term moves you the right *cube*, the second to the proper plane, the third picks the row and the last gets the column. It should be obvious that you can store multidimensional arrays column-wise also, with the leftmost index changing fastest. The equations for *column-wise* arrays are extended in a similar manner.

As you can see, the amount of storage in a four-dimensional array is the *product* of its four dimensions, so it's not likely we will run into useful arrays that are much bigger than this, unless each dimension is very small. In any case, if your interpreter can handle a single dimension, then the formulae above will allow us to work with as many dimensions as our memory can afford.

There are times when you might want to use a single-dimensional array in BASIC, even though you are working with two-dimensional data. For example, in order to set all the entries in a four-by-five data array to zero, we would normally use the statements shown in Fig. 2a, but it would take two less statements to use the routine in Fig. 2b. Since most BASIC interpreters don't let you refer to the same array with either one or two indices, we have to pick one way and stick with it.

I'm suggesting that you consider using the second way when most of your operations are with the entire array, and then use the formula given above when you need to refer to a particular point inside the array. This also applies if most of your operations are with whole rows of the array. You can set the array up by rows and operate directly on them. To set the second row to 2, you would use Fig. 3a or, if you were working on row I, where I is a variable, you would use Fig. 3b.

If you're working mostly with columns, you'll want to set up the array column-wise. Finally, if you're set up row-wise, but you want to run down a column, you can still do it without calculating each entry point. Suppose you want the second column set to X, then you can use the statements shown in Fig. 4.

The 2 is the important number since it picks out the right column, and if you're not sure what the last entry in the column is, just put in the last entry in the array—in this case 20—and it will still work. Most BASICs can STEP in FOR loops; if yours doesn't, you can set up a loop or compute each entry explicitly.

Ragged Tables

A ragged table is an array with a different number of columns in each row. You should realize that everything we say about ragged tables will also apply to column-wise information, so that we can concentrate on row-wise examples only.

```
100 FOR L=6 TO 10
110 LET A(L)=2
120 NEXT L
```

Fig. 3a. Setting the second row to 2.

```
100 LET M=(I-1)*5
110 FOR J=1 TO 5
120 LET L=M+J
130 LET A(L)=2
140 NEXT J
```

Fig. 3b. Setting row I to 2.

If we were dealing with a short table, we could just set the array dimensions wide enough to hold the longest row and proceed as before. However, one of the Irishman's many laws is that no matter how much memory space you have, you're bound to run into programs that need a couple more bytes of space.

Facing a ragged table and its empty space at the ends of most of its rows should give you ideas. In this case the trick is to keep track of where each row starts in another array, which we will call *pointers*, since they will *point* to each row. You don't get something for nothing, as you'll need a byte or so to store each pointer. So whether or not you can save space this way depends on how much total blank space you have at the end of your rows that are shorter than the longest row. Usually you will come out ahead.

Let's look at an example. Suppose we have a list of names that we want to keep in memory, as in a mailing list, for example. In a full array the names appear as in Fig. 5a, but in a ragged table we need only what is shown in Fig. 5b. To store this in memory, we make it look like Fig. 5c, where the second array contains the pointers to the start of each name in the data array, assuming each character is stored in a single location. You will see that the fifth pointer is needed to retrieve the last name.

To use the table we access it through the pointer array. The end of the name is found by subtracting one from the next pointer in the pointer array. For example, suppose we wanted to copy the third name in the table. We note that the third pointer is 14 and the fourth is 20. This means the third name is found in locations 14 through 19, which you can see is true.

Well, that's fine, you say, but I wanted my list in alphabetic order, and I was going to sort it before I used it, so it looks like I'll have to use a full array because I'll be shifting the names around from row to row. Nope. If we sort it, we aren't going to move the names around anyway; we'll just sort the pointers.

```
100 FOR L=2 TO 17 STEP 5
110 LET A(L)=X
120 NEXT L
```

Fig. 4. Setting the second column to X.

To do this we must understand that we are taking advantage of the position in the pointer array to find our data. To sort the data, all we need is another array that *points* to the pointers. We can't juggle the pointers themselves because we're using pairs of them to find the start and the end of our names. We could sort the pointers if we kept track of the start and end of each name, or the start and length of each name, but that would take another array of locations anyway, so we

can't save any space by doing it that way.

The array of pointers is shown in Fig. 6a. If we want to put KERN between JONES and LANCER, we sort the *position* array so that we end up with the arrangement shown in Fig. 6b. This isn't an article about sorting, so you can figure out your own way to do it. One way is to keep interchanging pairs until everything is in order.

Now that the list is in order, if we want the third entry, we go first to the position array, where we see that the third entry is a four. This tells us to look at the fourth position in the pointer array (and the one after it to get the end of the name); you'll find that we access 20, 21, 22 and 23 to obtain KERN from our list. Notice how the extra pointer helps make life simpler when we are accessing the pointers.

You might think you could save one more location in memory by dropping the one in the first pointer location, but the extra programming needed to check for it after you sort the position array is going to use more space than the one location you save. You may not need it if you aren't doing any sorting, but you might as well keep it, too, unless you are really desperate.

1	9	14	20	24	Pointers
1	2	3	4		Position in list

Fig. 6a. Original pointer array.

1	9	14	20	24	Pointers
1	2	4	3		Positions in list

Fig. 6b. Sorted pointer array.

Lists

In programming, a list is something more than the series of items you're going to buy at the supermarket. A list is a particular type of information structure. The difference between a list and an array is that you can add and remove items from a list without moving the old items, while still keeping everything in order.

If we had a shopping list in alphabetical order and we wanted to add *eggs* between *donuts* and *frankfurters*, we would probably copy the list over. In memory, this is what we would do if our items were stored in an array, moving everything down after *frankfurters* so that we could stick in *eggs*.

If our data were stored in a sequential access memory—such as a cassette tape—we would be forced to do this. However, since we are working with random access memory here, we can set up our data in another way. The basic idea is related to what we did when we were sorting our ragged tables. In this case we will keep a pointer with every data entry.

Our purpose here is not so much to save space in storing the actual data entries, but to make it easy to change the data. This should shorten our programs that use the data, conserving space in another way. We will also see an example where a list structure will reduce the amount of storage we need directly. Lists are usually called *linked-lists* to distinguish them from arrays. There is almost always some kind of order or direction associated with a list.

A simple example should help. In Fig. 7 each box is a location in memory, and the adjacent box is an adjacent location in memory that contains a *pointer* to the next item in the list. The pointers are the links in the list. Traditionally, we use a *ground* symbol to indicate the end of the list. The data kept in the pointer box is just an address in memory or a location in an array.

For example, if B is in location 21, and D is in location 27, then the pointer that goes with B will be kept in location 22, and the value stored in it will be 27, the address of D. The data items are not next to each other because they don't have to be. They can be anywhere in the available space.

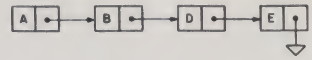


Fig. 7. A list with four data items.

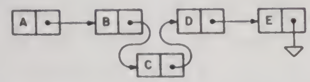


Fig. 8. List with item C added.

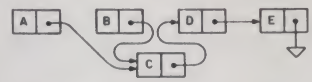


Fig. 9. The list with item B removed.

To add item C to this list while maintaining alphabetical order, we scan down the list until we find where it goes, move the pointer with B so that it points to C and reconnect the list by making C's pointer point to D (see Fig. 8). The items that were in the list haven't moved; in fact, the only thing that changed in the original data was the pointer next to B.

Suppose we wanted to remove B from the list. All we do is scan down until we find the pointer that points to B and change it so that it points to what B's pointer is pointing to (see Fig. 9). You can see that we didn't have to throw B away, nor did we have to bother with its pointer.

This works fine as long as (1) we have plenty of space, (2) we have a way to recover space later or (3) we never remove items from our list. Usually this is not the case, so we will want to keep track of our available space, which will also be organized as a list. When we drop B we have to put its space back into our list of available space. This means our memory will be divided into two lists at all times—one for the data items and one for the space that will be available to expand and contract our data list.

H	A	R	R	I	S	O	N
J	O	N	E	S			
L	A	N	C	E	R		
K	E	R	N				

Fig. 5a. Complete table.

H	A	R	R	I	S	O	N
J	O	N	E	S			
L	A	N	C	E	R		
K	E	R	N				

Fig. 5b. Ragged table.

Data:	H	A	R	R	I	S	O	N	J	O	N	E	S	L	A	N	C	E	R	K	E	R	N		
	1				5					10					15					20					25

Pointers:	1	9	14	20	24

Fig. 5c. Ragged table in memory with pointer array.

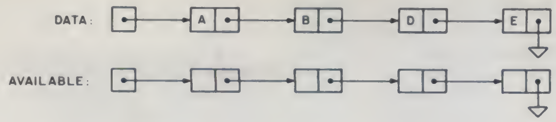


Fig. 10a. The data list and available space list.

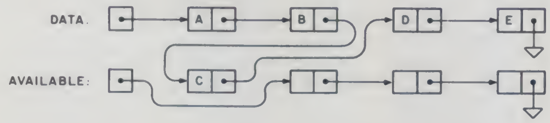


Fig. 10b. The lists after item C is added in its place.

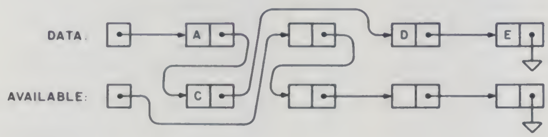


Fig. 10c. The lists after item B is removed.

We need to keep track of where each list starts, so we store a pointer to the head of each list, called a *header*. If our total space in memory was only enough for eight data items, plus the headers, our two lists would appear as shown in Fig. 10a at the start. When we added C it would change to Fig. 10b, and when we removed B it would change to Fig. 10c.

The diagrams show graphically what is happening to the lists, but to be more exact, we need to develop some routines that operate on lists. First, we need to think about what a list looks like in memory, rather than in the simple diagrams above. In BASIC, a list would be kept in an array.

We'll put both the data list and the available list in array L, which will be made big enough to hold all the data that will be present at any one time. The data will go into odd locations, and the pointers into even locations. We'll use L(1) for the header that stores the address of the first piece of data, L(2) for the available list header and a minus one for the ground symbol that shows we are at the end of the list.

At the start of the program, our list will probably contain no data, and all locations will be available. We initialize the list with the routine shown in Fig. 11. The L(1) data header is grounded, since there is no data in the list yet, and the available storage is strung together so that each pointer is pointing to its neighbor.

The next step will be to add some data to the list and assume for the moment that the data is arriving in order, so that all we have to do is find the end of the list and put the data there. To do this, we first check L(1) to see if it is empty, and if it is we quit searching. Otherwise, it will be pointing to a location in the array. We then check the pointer with the next data location to see if it is grounded and then repeat this until we find the end of the list. Fig. 12 illustrates the BASIC routine to find the end of the list.

You should make sure you understand what is going on in Fig. 12 since we will be using similar techniques in all our list-processing routines. Statement 110 initializes, statement 120 tests, and exists if the test is met. If the test fails, then L(E) holds a pointer that is the address of the next piece of data in the list. Statement 130 computes the address of the *pointer* that goes with this next piece of data and then transfers the value of this pointer back into E, so that we can return up to statement 120 to repeat the test and eventually find our way out to statement 150.

It's important to remember that when you know the address of a piece of data, the address of the next piece of data in the list will be found right next to the location of the first piece, and that all you have to do to get the pointer for it is to add one to the first location number.

Now that we have found the tail end of the data list, we add our new data to the list. To do this, we remove the first available space from the available list and put it into the data list, being *careful to update the* pointers in both lists. This is shown in Fig. 13.

If your BASIC can handle computation within subscripts or subscripted subcripts, you can condense the program above. At the end of the routine, E still contains the address of the end of the data list, so you could repeat this routine immediately if you were adding more than one data item to the list.

Next we need a routine to remove an item from the data list and recover the storage location for the available list. To do this, we must know the location of the pointer that points to the item we are going to remove. Let's assume it's in R. Fig. 14 shows the routine in BASIC. As before, R is set to the location containing the pointer to the next item on the list, to make it easy to remove the following item, too.

Finally, we need a routine to insert an item into the data list. It is like the first routine, except there is another pointer to take care of. Assume that I is the lo-

```
10 DIM L(18)
15 REM GROUND THE DATA HEADER
20 LET L(1) = -1
25 REM SET THE AVAIL HEADER TO FIRST AVAIL LOCATION
30 LET L(2) = 3
35 REM CONNECT THE AVAIL LIST
40 FOR I = 4 TO 16 STEP 2
50 LET L(I) = I + 1
60 NEXT I
65 REM GROUND THE AVAIL LIST
70 LET L(18) = -1
75 REM FROM HERE GO TO FIRST PROGRAM STATEMENT
```

Fig. 11. List initialization routine.

```
100 REM SUBROUTINE TO FIND END OF LIST
110 LET E = 1
120 IF L(E) = -1 THEN 150
130 LET E = L(E) + 1
140 GO TO 120
150 REM EXIT HERE WITH E SET TO LOCATION OF END OF LIST
160 RETURN
```

Fig. 12. Routine to find end of list.

```
200 REM SUBROUTINE TO ADD DATA TO LIST AT E
201 REM DATA TO BE STORED IS IN VARIABLE 'D'
205 REM STORE THE DATA
210 LET P = L(2)
220 LET L(P) = D
225 REM CONNECT THE NEW DATA TO THE DATA LIST
230 LET L(E) = P
235 REM SAVE THE POINTER
240 LET E = P + 1
245 REM RESTORE THE AVAIL HEADER
250 LET L(2) = L(E)
255 REM GROUND THE DATA LIST
260 LET L(E) = -1
270 RETURN
```

Fig. 13. Routine to add data to the list.

```
300 REM SUBROUTINE TO REMOVE AN ITEM FROM LIST
301 REM USES R FOR THE ADDRESS OF THE DATA ITEM
302 REM USES T AS A TEMPORARY VARIABLE
305 REM REMOVE THE DATA ITEM & STORE IT IN D
310 LET P = L(R)
320 LET D = L(P)
325 REM MOVE THE POINTER FROM THE DATA BACK TO R
330 LET T = P + 1
340 LET L(P) = L(T)
345 REM CONNECT THE VACANT SPOT TO THE AVAIL LIST
350 LET L(T) = L(2)
355 REM RESTORE THE AVAIL HEADER
360 LET L(2) = P
370 RETURN
```

Fig. 14. Routine to remove an item from a list.

```
400 REM SUBROUTINE TO INSERT ITEM IN LIST
401 REM USES I FOR THE POINTER WHICH WILL CHANGE
402 REM USES T AS A TEMPORARY VARIABLE
405 REM STORE DATA IN FIRST AVAIL SPACE
410 LET P = L(2)
420 LET L(P) = D
425 REM SAVE THE POINTER IN POSITION I
430 LET T = L(I)
435 REM CONNECT THE FRONT OF THE DATA LIST TO THE NEW DATA
440 LET L(I) = P
445 REM RESET THE POINTER
450 LET I = P + 1
455 REM RESTORE THE AVAIL HEADER
460 LET L(2) = L(I)
465 REM CONNECT NEW DATA TO THE TAIL OF THE DATA LIST
470 LET L(I) = T
480 RETURN
```

Fig. 15. Routine to insert an item in the list.

```
500 REM SUBROUTINE TO FIND AN ITEM IN THE LIST
501 REM SEARCHING FOR ITEM 'D', STORES POINTER FOUND IN P
502 REM USES T AS A TEMPORARY VARIABLE
503 REM USES F AS A FLAG TO SHOW SUCCESSFUL SEARCH
510 LET F = 1
520 LET P = 1
525 REM STORE NEXT POINTER IN T
530 LET T = L(P)
535 REM CHECK T FOR END OF LIST
540 IF T = -1 THEN 580
545 REM TEST FOR ITEM
550 IF L(T) = D THEN 590
555 REM GET NEXT POINTER & TRY AGAIN
560 LET P = T + 1
570 GO TO 530
580 LET F = 0
585 REM SET F TO 0 BEFORE EXITING IF END OF LIST HIT BEFORE
586 REM ITEM FOUND, OTHERWISE EXIT WITH F STILL SET TO 1
587 REM AND P SET TO LOCATION OF POINTER DESIRED
590 RETURN
```

Fig. 16. Routine to find an item in the list.

cation of the pointer to the data item that is going to *follow* the one we want to insert. The routine we use is shown in Fig. 15. As in the other routines, the variable I is left set to the location of the pointer to the next data item, which makes it easy to insert more than one item at a time into the list.

Would this routine work if we were inserting at the end of the list? The answer is yes. So we really don't need the first routine, unless all our insertions will be at the end of the list—which sometimes happens—as in a FIFO (first-in/first-out) queue, for example.

This brings out an important point, namely, that the most frequent cause of programming errors when we're working with lists seems to be that we forget that lists can be empty, and the pointer that we expect to be pointing to a data location might actually contain a *ground,* so that when we access the array, we find ourselves jumping outside the array entirely. This happens when we run out of space in the available list. In the insertion routine, for example, we should test P after statement 410 to see if it is a ground and jump to an error message if it is.

The only thing we need now is a routine to find items in the list. You can see that it is the same sort of thing we used to find the end of the list, except the test is made on the data item instead of the pointer, and the pointer to the data item is what we save.

Suppose we're looking for a D, then in BASIC we would use the routine in Fig. 16. You should notice what happens if the item isn't found in the list. A good exercise is to rewrite the routine so that it prints every item on the list, since it will come in handy when you start debugging.

Other Kinds of Lists

There are many different kinds of lists. The ones we've been looking at are the simplest. Sometimes it is necessary to be able to scan the list in both directions. To do this we keep a *back* pointer as well as a *forward* pointer with each data item. Inserting and removing items is about the same, except that we have another set of pointers to keep track of, and there is a *header* at each end of the list to maintain. This is called a *doubly linked* list. The diagram form is shown in Fig. 17.

Another type is the *circular* list, which has no end. The pointer that normally would be grounded is made to point to the front of the list. Once you start scanning a circular list you'll keep going, so you had better be sure that the item you're looking for is on the list.

Instead of a header, you could keep a pointer set to your current position in the list. Alternately, you can keep a header as a member of the list, so that the list will never be empty, and you won't need special programming to handle the case where the list disappears. This also solves the problem of scanning for an item that you're not sure is on the list, since you can start and stop at the header. Fig. 18 shows a circular list.

If you have more than one list at work, you'll probably put them in the same array, drawing on the same available list. Simulation routines are heavy users of lists, and they will typically keep a list for each queue that forms but operate with a single available space list. The timing routine, which controls what happens next in a simulation, is another common application for a linked list.

Our ability to point to data that is used in several places can save us space, just as the use of subroutines avoids repeating parts of a program. The other way to save memory is to skip over what you don't need.

This brings us to an example of a situation where you can save space with a list. Suppose you had a rather large array with only a few entries in it. Instead of saving the entire array, you could just keep a list of each entry and where it was located in the array. If all the items are the same, or mostly the same, it makes things

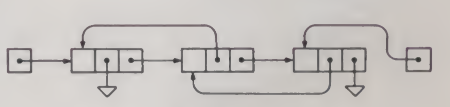


Fig. 17. A doubly linked list.

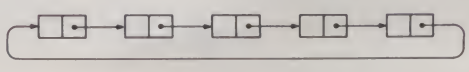


Fig. 18. A circular list.

even simpler.

Assume, for example, that you had a 100 × 100 array that contained nearly all zeros, but had a few ones scattered around it. To store this directly in memory as an array will require more storage space than most of us can afford, but it still may be possible to store it as a list. First, we would create a list that showed the rows that had ones in them, and then a list for each row that shows which columns were present in the array. The structure is shown in Fig. 19, which shows that row three, column 16 (3,16) has a one in it and also positions (7,7), (7,32), (7,33) and (91,4).

You can see that the vertical list of each row data element contains the header information for the column data, along with a pointer to the next piece of row data. The column elements contain just the column number and a pointer to the next column element.

So far, we have used only 20 storage locations instead of the 10,000 required to hold the full array. It should be obvious that this only works when you have a very sparsely populated data array. In the worst case, you would need a row header for each column element, or five locations to store enough information to find each number in the array, and you'll come out ahead as long as 80 percent of your array is empty.

If the data entries are different, then it takes as many as six locations for each number in the array, since we need to keep the data value itself along with the column number and the pointer kept in the column lists. For big arrays with few entries, this will still be worthwhile.

Related Techniques

Linked lists have other relatives, such as stacks, queues and trees. The techniques used in working with them are much the same.

Storing the array above by rows and then by columns is really an example of a simple tree structure. We could have stored the array in a simple list by using our formula for computing the index position and then doing all our operations with the array via the formula. The tree makes it a little easier to get at the values in the array, since you may avoid scanning some of them when you are looking for something.

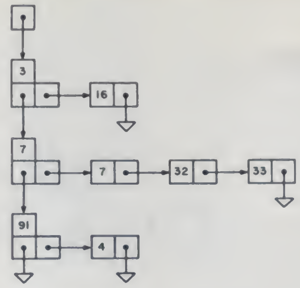


Fig. 19. A large array stored as a linked list.

Depending on how you plan to use the array, you may want to set it up differently than is shown above. For example, if the entries are different and you will be scanning the whole array when you are looking for things, then it helps to have the column lists linked back to the row headers as in a circular list, or perhaps to the *next* row header so as to link every element in the array in sequence.

Another way would be to store each value along with its row and column numbers and a pointer to the next value, which only takes four locations per entry instead of six and would be a list instead of a tree. It would make scanning the whole thing easier but scanning a single row much harder, so again, how you set it up depends on how you're going to use it.

As you can see, there are many possibilities. If you would like to investigate further, check your local college library for a copy of Donald E. Knuth's *The Art of Computer Programming, Volume 1, Fundamental Algorithms* (Chapter 2—"Information Structures"), Addison-Wesley, Reading MA, 1968. Besides material on other information structures, there is a wealth of details on arrays, tables and lists. ■

✓ Reader Service—see page 179

Ronald G. Parsons
9001 Laurel Grove Dr.
Austin TX 78758

Disk Power!

Which disk is for you? That's a tough question, and it doesn't have any simple answers. Here's a vote for Processor Technology's Helios II and PTDOS.

Making a selection for a floppy-disk system is definitely not an easy thing to do . . . there is a multitude of parameters to consider. Needless to say, the reason we bring you articles such as the following is to help make that selection process a little easier. If you're really serious about doing an extensive comparison of standard-sized disk drives, controllers and software, let us recommend that you part with $14 and invest in a copy of Selecting A Floppy Disk System *from Center for the Study of the Future, 4110 N.E. Alameda, Portland OR 97212. Very comprehensive and very thorough. It should definitely be available, either as reference or on the bookshelves, in every computer store in the country.—Eds.*

Just a few more changes to that long assembly-language program and it will be done. Load the tape into the cassette player; wait a seemingly interminable five minutes while the source file and the assembler are read in. Make the changes. Now assemble it . . . good, no errors. The changes were only minor, so let's try running it—then we'll save it back on tape. Execute . . . Zap! . . . there went all of memory, overwritten by a program run amok. Five more minutes to read in the source and assembler files again.

Well, it finally happened! I made the decision to get a floppy-disk system for my 8080 system—a SOL-20. But which one? . . . What size—full or mini? . . . What manufacturer's drive? . . . But most important—what system software is available or furnished? A few quick calculations of the number of bytes I wanted available on line showed I needed a full-size floppy-disk system. Two drives would be needed so backup copies of the disks could be made.

The National Computer Conference in Dallas in June 1977 had a large personal-computing exhibit that afforded visitors the opportunity to inspect several different systems. An advance copy of the external specifications of Processor Technology's disk-operating system, PTDOS, was given to me by Gary Ingrahm, Processor Tech's president. It seemed to have all the features I needed. Their disk hardware, the Helios II Disk Memory System, was on display, but not yet available for delivery. The only other disk-operating system (DOS) that seemed to come close to meeting my needs was CP/M from Digital Research. Comparisons of CP/M and PTDOS are made throughout this article.

The Helios II system and PTDOS came out on top after my evaluation. In addition, PTDOS had a clean interface to SOLOS, the operating system on my SOL-20 and its built-in video display system.

In the following sections, I will describe the Helios II system and PTDOS, and present some examples of uses of the system.

The Helios Hardware

The Helios II Disk Memory System consists of a PerSci 270 dual, full-size floppy-disk drive, a cabinet containing a power supply and an indicator panel, a controller board and a formatter board. The controller board plugs into the S-100 bus and is connected to the drive with a six-foot ribbon cable. The formatter board can be plugged into the S-100 bus but obtains only power from the bus. The formatter and controller are connected to each other by another ribbon cable.

The PerSci dual drive is unique in several ways: Both drives use a single motor to spin the diskettes; they also use a common head-positioning linear, or voice-coil, motor. The use of a single drive motor and axle results in a dual drive no larger than most single drives.

The linear motor is used to position the read-write head to the different tracks on the diskette. Most floppy drives use a stepping motor and a drive screw to move the head from track to track. It takes from five to ten ms to move the head in or out one track. Thus, up to 760 ms may be required to position the head from track zero to track 76. The PerSci voice-coil head positioner can move a single track in ten ms and can move from one track to any other in no more than 100 ms.

The average access time to find an arbitrary sector on a diskette is about 116 ms compared to about 250 ms for most other drives. The PerSci drive does not require cooling since the average operating power is only 28 Watts.

The cabinet of the Helios II encloses the PerSci drive and contains a power supply for the drive, a fan to provide filtered, positive-pressure air circulation to help keep dust out of the drive, and an indicator panel with LED lights to show the status of the drive. The cabinet has enough room (and mounting holes) for a second PerSci 270 dual drive. An intriguing assortment of back-panel cutouts hints at the future possibility of adding a backplane to the cabinet for expansion or an all-in-one-cabinet computer and disk system.

The controller board is the interface between the S-100 bus (and the rest of the computer) and the disk drive. The controller board runs very hot and requires additional forced-air cooling. The functions of the controller include translating OUT instructions from the software into control signals for the disk drive, providing status signals available by using an IN instruction, buffering data to and from the disk and memory and controlling DMA (more on DMA later).

The formatter board creates the necessary timing signals to maintain the required format of the information on the disk. Each sector's data must be accurately positioned with respect to the sector timing holes on the diskette.

Besides the data in each sector, other information is present to indicate the amount of data in the sector, the location of the previous and next sectors, CRC data (cyclical redundancy check, an error-detection scheme), etc. The formatting of

The author's Helios/SOL System (Photo by Jane Steig Parsons).

this information is handled by the formatter board. All signals to and from the formatter go to the controller board. The formatter board may occupy a slot in the S-100 bus backplane, but, if so, it obtains only +8 V power from the bus. A separate connector is provided to supply power to the formatter if it is not placed in the backplane.

When data is to be transferred to or from the disk, the controller requests control of the bus by bringing the bus line PHOLD to a logic-low state. When the 8080 processor is finished with the instruction in progress, it suspends operation and brings the hold acknowledge line, PHLDA, high. As long as PHOLD is held low, all processing by the 8080 will be suspended. The controller has control of the bus and transfers data directly to or from memory without involving the processor. This process is called DMA (direct memory access).

DMA is done in bursts of about 20 μs moving about 12 bytes. The processor then regains control for about 370 μs. This process repeats until an entire block is transferred. The entire DMA transfer may last for as long as 130 ms depending on the block length. The transformation to and from bit parallel (bus) to bit serial (disk) is done on the controller by a FIFO buffer.

During DMA, the controller must place signals on the bus so memory devices will function properly. Usually this is not a problem with static memory boards since they need only a few bus signals, besides the address and data lines, to function properly. Dynamic memory boards can be a different story. Many manufacturers of dynamic memory boards or DMA controllers make their own assumptions about the bus signals.

Each memory location in dynamic memory must be refreshed (read or written) every millisecond or so. If it is not, memory contents may be lost. A dynamic memory board monitors the bus signals and chooses times to do refresh when it will not interfere with other memory access on the board. If, during DMA, the controller does not place the expected signals on the bus, refresh may not occur. Not all dynamic memory boards will work with DMA. If you're going to use dynamic memory, be sure to test it on your system before buying the memory.

PerSci has two options for the Model 270 drive that are not included in the one furnished with the Helios. One option allows the controller to eject either diskette by remote control; the other provides a sensor to detect whether the write-protect notch is covered on the diskette inserted in the drive. Write protection is available under software control, but only on an individual file or on the entire system. It would be nice to be able to write-protect an individual diskette.

The PTDOS Software

PTDOS is a complex, easy-to-use, sophisticated, file-oriented operating system provided with the Helios II system. The system requires a minimum of 12K bytes of memory from 9000 to BFFF hex for the resident portion of the DOS, including 8C8 hex bytes for file buffers. Additional memory is required at 100 hex to execute certain commands. A minimum usable configuration would probably be 8K at 0 and 12K at 9000 hex. Additional memory at 0 would allow larger work space for editors, assemblers, BASIC, etc., while an additional 2K to 4K just below 9000 hex *would* provide additional space for file buffers or space for device drivers.

Fig. 1 compares these memory requirements to those for CP/M. A 16K CP/M system has a user transient area of 2800 hex bytes and has a single 80-hex-byte file buffer. A 16K PTDOS system has a user transient area of 1000 hex bytes, but 8C8 hex bytes available for file buffers. A user transient area is memory available to the user for executing certain commands and for storing and running programs.

Thus, for equivalent buffering, PTDOS requires about 1000 hex (4K) more memory than CP/M. As we shall see later, this additional memory brings with it additional function and capability. To understand the file buffer requirements, we must understand the file block structure, which will be discussed later.

Most disk operating systems for hobbyist computers provide user commands by which the user can execute and manipulate files. Some elementary systems provide eight to ten simple commands executed by a single letter such as K for killing or deleting a file from the disk. More complete systems provide a lot of commands, each called by an English word or abbreviation. A summary of the commands provided by CP/M and PTDOS is given in Fig. 2. These commands are usually typed on the console and may be in either uppercase or lowercase.

In PTDOS, each file is given a name of one to eight characters which must be unique on a diskette. If a diskette is inserted in a drive other than the default unit (usually unit zero), a slash and the unit number must be appended to the file name, e.g., LETTER/1 or MEMO/1.

As an example of a PTDOS command, the COPY command creates a copy of an existing file.

COPY FSDISP,*FSDISP/1*

takes the contents of the file FSDISP on the default unit (say unit zero) and copies them to a file of the same name on unit one. If a file of that name does not exist on unit one, a new file will be created. If such a file does exist, the old contents of the file will be replaced.

COPY TEMP,TESTFILE

copies the current contents of file TEMP to the file TESTFILE on the same unit.

A file may be deleted (erased) from a diskette by the PTDOS command KILL.

KILL TEMP,FSDISP/1

deletes the files TEMP (on unit zero) and FSDISP (on unit one). The space on the diskette formerly occupied by these files is made available for later reuse by other files.

Most PTDOS commands perform a complete operation on a file or set of files. In other words, a file may be created if nonexistent, opened (made ready for processing), read or written, end-filed (the physical end of the file is marked and any following information is discarded), and closed (buffers emptied and the file name removed from the list of active files). An operating system primitive, on the other hand, is a function provided by the operating system that can be called from an assembly-language program. A summary of the operating system primitives available with CP/M and PTDOS is given in Fig. 3.

A command (file name) typed on the console causes a file of that name to be read into memory and, perhaps, executed. The executing program calls on a series of operating system primitives to perform its complete task. A user-written program working with disk files also makes operating system primitive calls. In fact, the only difference between a standard PTDOS command and a user-written program is the person who writes the program.

Should an error occur, such as an attempt to write on a write-protected file or a request to open a nonexistent file, a jump is made to a user-written routine—this will process the error.

There are three levels of errors: very serious (disk surface bad, controller error, etc.), moderate (out of memory, nonexistent file) and warning (end-of-file). The operating system is normally set to handle very serious and moderate-level errors; the user program handles warning-level errors. The user program can, however, set a system parameter to allow the program to handle either or both the very serious or moderate-level errors. A utility program is provided to print English text error messages or user-provided text when the error return code is passed to the utility.

Besides the PTDOS operating system itself, several other programs are provided on the same diskette. There are two text editors for creating and modifying text files, such as assembler source code or BASIC programs. One of the editors is video oriented, and a 16-line page of a file is displayed on the screen. The file can be scrolled forward and backward a line or a page at a time. The text can be modified under cursor control. The usual features such as character (or line) add, change or delete, as well as string search and block move, are provided. The other editor may be used with any output device.

An assembler that assembles source files from disk and places the object code back on a disk file is provided. Memory is used only to hold the symbol table. A COPY pseudo-instruction is available in the assembler to copy source files into the input stream of the assembler at the point of the COPY command. With this instruction, several files can be assembled together. Conditional statements can be included in the source so that sections of code can be included or excluded based on the values of specified variables.

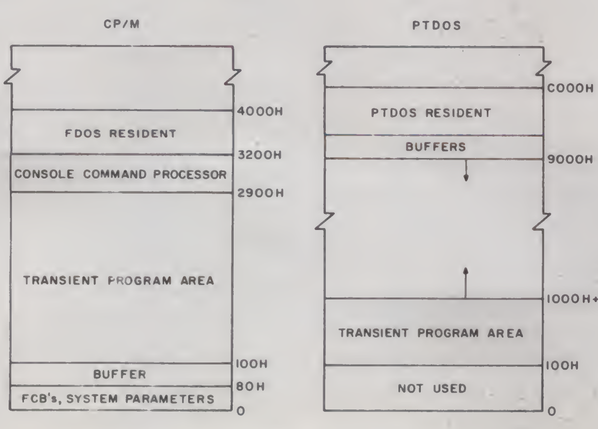


Fig. 1. Memory utilization by CP/M and PTDOS.

Two higher-level languages are provided—Disk BASIC/5 and FOCAL. The BASIC is similar to Processor Technology's cassette BASIC/5 but without the on-screen editing feature or the ability to read or write data files. Source files can be saved or retrieved from the disk. BASIC/5 on the disk is a limited version of BASIC, so don't expect to do much serious work with it; you'll have to get their Extended Disk BASIC to do that.

A dynamic debugger is included—in two identical versions, except that one runs at 3000 hex and the other at 5000 hex. Breakpoints can be set and registers inspected and modified while your program being debugged is running. Memory can be inspected or modified in hex, character or instruction format. Dumping memory in instruction format produces disassembled code.

A macro command facility is provided. This allows the user to set up a file containing an often used series of commands. For example, suppose we create a file named STATUS and the file contains:

FREE?
NFILES?
FSDISP
$STOP

Giving the command DO STATUS causes each of the files, FREE?, NFILES?, and FSDISP, to be loaded and executed. The line $STOP returns control to the normal input mode on the console.

Of course, Processor Tech's Star Trek game, TREK80, is included as one of the programs on the system disk.

The Documentation

The documentation for the hardware was quite complete except for the "Theory of Operation" chapter. As originally received, the chapter consisted of two charts of signal lines. A phone call revealed that the chapter would be provided later. (Why is documentation always the poor cousin of computer hardware and software?)

I've seen an advance copy of that chapter and it is very complete—down to an explanation of nearly every gate and logic block. The schematic diagram for the controller suffers from having too much on a single page. I would have preferred several sheets, each containing a logical section of the controller, much like the schematics for the SOL computer.

The documentation for PTDOS is also good but could certainly use an extensive index. Just about everything is there, although sometimes it's a little hard to find. Once the documentation is understood, the commands and system calls are quite easy to use.

PTDOS Files

All user information on a PTDOS disk is organized in files. To the user, a file is just a sequence of bytes having a beginning and an end. When a file is created, a name is assigned by the user to the file. Certain other characteristics may be assigned by the user at that time or the system will assign a set of default characteristics. For a PTDOS file, these characteristics include a file type, a blocksize, and a set of protection attributes, besides the file name.

The file name consists of one to eight letters, numbers, and certain special characters such as $, : or ?. The file name must be unique on a diskette. The file type is a single character optionally preceded by the letter I, which signifies that the file is executable code. Other than that, the file type is arbitrary but can be used to classify files—T for text files, S for source files, $ for DO files, etc.

the $988 Surprise . . .

If you haven't looked carefully at the Level-II 16K TRS-80, you're in for a big surprise! Level-II BASIC gives TRS-80 advanced features like comprehensive string handling, multi-dimension arrays, multi-letter variable names, named cassette files, full editing, integer arithmetic, single (6-digit) and double (16-digit) precision arithmetic, formatted printing, memory-mapped video (print directly at any of 1024 screen positions), 128x48 video graphics (may be intermixed with text), error trapping, auto line numbering, TRACE, PEEK and POKE . . . to name just a few. Because Level-II is in ROM, TRS-80 powers-up ready to go with the full 16K RAM available for your use.

This means TRS-80's memory is equivalent to a 28K RAM-based system.

New for 1979—TRS-80's numeric (calculator) keypad included on every 16K computer, and available as an add-on for present owners.

TRS-80's modular design allows easy expansion. Add up to 48K RAM, Expansion Interface, printers, 1 to 4 Mini-Disks, RS232C, telephone acoustic couplers, Voice Synthesizer, dual cassette recorders, our System Desk and Printer Stand. Surprisingly, these are not promises of things to come, but real products being delivered right now. Software from games to General Ledger are available, with more cassette and disk software being added monthly.

Radio Shack's 58 years of consumer electronics leadership, our 50 regional repair centers (growing to 100 this year), our new Radio Shack computer centers, and our NYSE-listed billion-dollar parent, Tandy Corporation, insure that customer support is always available right where it should be—locally.

So if you haven't seriously looked at TRS-80 yet, ask your local Radio Shack for our new 20-page fact-filled catalog and be prepared for a $988 surprise. Surprising power—features—price—support! Level-II 16K systems include everything pictured, plus the manual. Better to be surprised now . . . before you choose the wrong microcomputer system.

Reader Service—see page 179

CP/M	PTDOS	Comments
ASM file[,parms]	ASSM ifile,[lfile],[bfile],[S=opt],[efile][,sfile]	See text
----	* BLDUTIL file[,parms]	Add or delete utility files on file
----	* CLOSE #n[,#m]...	Close indicated files (See OPEN)
----	CONFIGR [/u,]password	Change or inspect disk parameters
PIP [command line]	COPY ifile,ofile[,parms]	Copy and/or concatenate files
	O=ofile,ifile,...	
----	* CREATE file[,filetype[,blocksize]]	Create file with given type and blocksize
SYSGEN	DISKCOPY /from,/to	Copy diskette contents
DUMP file	DUMP file[,from-addr[,to-addr]]	Dumps file in hex (PTDOS also gives ASCII)
ED file	EDIT ifile[,ofile]	Text editors
	EDT3	
----	* ENDF #n[,#m]...	Put end file at current position (See SPACE and OPEN)
----	* EXEC addr[,parms]	Execute program at addr. Program may read parms
----	EXTRACT file[,parms]	Display image file block length
* DIR file	FILES parms	List files from directory
STAT	* FREE?	Prints the amount of space remaining on disk
----	GET parms	Retrieves files saved by SAVE
* SAVE nblocks file	* IMAGE file,addr-range[,start-addr]	Write a file containing an image of memory
* ERA file	* KILL file[,file]...	Removes files from disk
----	* OPEN file[,parms]	Opens a file for processing
----	* OPEN?	Lists all open files by number
----	* OUT V or OUT P	Sets output drivers to video (V) or serial (P)
* TYPE file	PRINT parms	Prints ASCII file
----	RANDOM file	Creates an index for random accessing (not needed for CP/M)
----	* READ file[,addr-range]	Transfer a file to memory
----	* REATR file[,parms]	Change the file's attributes
----	RECOVER [/u]	Recover "lost" sectors on disk after errors
* REN new=old	* RENAME old,new[,old,new]...	Rename old file to new name
----	* RETYPE file,type	Change file's type
----	RNUM file[,parms]	Renumber a line-numbered file
----	SAVE parms	Archive files to a save file
----	* SEEK file,parms	Position an indexed file to specified byte or block
----	* SET parms	Specifies various system parameters
----	* SETIN file	Set input file for Command Interpreter
----	* SETOUT file	Set output file for Command Interpreter
----	* SPACE file,parms	Position file forward or backward
----	SYST parms	List system parameters from disk or memory
----	* WRITE file,addr-range	Transfer memory to file
----	ZIP hex-number	Sets memory below system to specified value
LOAD file	----	Converts Intel hex file output of assembler to binary
SUBMIT file,[parms]	DO file[,parms]	Execute commands on file with parameter replacement
DDT	DEBUG	Dynamic debugging program

* Commands do not execute in or disturb transient area.

Fig. 2. Summary of user commands available with CP/M and PTDOS.

The blocksize of the file determines the size of the physical blocks of a file on the disk but does not in any way affect the logical structure of the file. The blocksize can be from one to 4095 bytes. Certain sizes produce more efficient use of the diskette surface.

The protection attributes can be set by the user so the file can or cannot be read, written, killed, displayed by the FILES command, have its name changed or have its disk structure changed.

The way a PTDOS file is stored on the diskette is unusual. The usual hard-sectored disk format has one block of 128 data bytes in each sector. The beginning of each sector is determined by one of the 32 sector holes on the diskette itself. The Helios formatter ignores every other sector hole so that there are 16 double sectors on each of the 77 tracks.

A block one sector in length contains 256 bytes of data. Of course, each block has, in addition, control bytes used only by the operating system. If the blocksize of the file is greater than 256 bytes, the format of the file is extended by writing in the space between sectors so that the physical blocks are longer than one sector. By using two sectors, enough space is saved on the track to add 64 bytes between the sectors so that two sectors can hold 256 + 64 + 256 = 576 bytes.

For each additional sector, an additional 64 bytes is gained. A file with blocksize of 4095 requires 13 sectors, saving 767 bytes. The remaining three sectors on the track could be used for a file of blocksize 896 (380 hex). If every file had a blocksize of 4095 or 896 (which would not be practical), the theoretical capacity of the diskette would be 77 x (4095 + 896) = 384,307 bytes. A diskette formatted in the standard way could contain 77 x (32 x 128) = 315,392 bytes. A soft-sectored diskette has a capacity of 256,256 bytes. The combination of multiple sectors and user-defined blocksize increases the capacity of the diskette over that provided by the standard format.

This method of formatting is called "firm" sectoring by Processor Technology since it uses a combination of techniques used by hard-sectored and soft-sectored disk formatters. The disadvantage is that the diskettes written by PTDOS and Helios are usable only on other Helios systems. The disk format written by CP/M can be the standard format of 77 tracks with 26 sectors each containing 128 bytes.

Will the nonstandard format be a hindrance? Only if you wish to get disk files from others or trade files with non-Helios users. Perhaps a significant amount of software will become available on PTDOS diskettes. At present, CP/M is almost a de facto standard. You must answer for yourself whether this is a problem.

Using PTDOS

PTDOS is bootstrapped into memory by a short (51 hex byte) bootstrap program. This bootstrap can be loaded into memory from cassette or, better, put into ROM in your system. The bootstrap program loads a file from disk track zero, sector zero, into memory and executes it. This new file then loads the resident part of PTDOS into its memory space (9000 to BFFF hex) and transfers control to the command interpreter.

In this process, a file named START.UP is executed as a DO file. As supplied, the PTDOS diskette contains a START.UP file that displays the system parameters (disk name, disk date, space available for buffers, etc.). The user may add, change or delete items on this file as he sees fit. For example, operating instructions could be placed on this file to help inexperienced users get started.

After bootstrapping the operating system, the next step is creating and editing a file. There are two text editors on the system disk. The one I

Reader Service—see page 179

CP/M	PTDOS	Function	Comments
yes	yes	Read console	ASCII character
yes	yes	Write console	ASCII character
yes	user	Read reader	PTDOS: user written utility
yes	user	Write punch	PTDOS: user written utility
no	source	Read cassette	PTDOS: source provided
no	source	Write cassette	PTDOS: source provided
yes	no	Read string	ASCII string
yes	no	Write string	ASCII string
yes	yes	Character available	Console input available
yes	yes	Create file	Allows specification of file type and blocksize (PTDOS only)
yes	yes	Open file	CP/M: address of file control block (FCB) must be passed PTDOS: file name passed
yes	yes	Close file	
no	yes	Close all	Close all open files
yes	not needed	Search for file	Fill FCB with information for named file
yes	not needed	Search for next	Fill FCB for next file matching file name
yes	yes	Delete file	Delete file from directory and free sectors for future reuse
yes	yes	Read block	CP/M: 128 bytes always read PTDOS: user specifies number of bytes read (may be 1 to 65535 independent of blocksize)
no	yes	Delimited read	Stop read when specified hex character found
yes	yes	Write block	CP/M: 128 bytes always written PTDOS: user specifies number of bytes written (may be 1 to 65535 independent of blocksize)
no	yes	Delimited write	Stop write when specified hex character is found
no	yes	Read single byte	
no	yes	Write single byte	
no	yes	Move file cursor	May be moved forward or backward. Disk I/O may take place
no	yes	Delimited move	May be moved forward or backward to specified hex character
no	yes	End file	Puts end file at current position and truncated all following data
yes	yes	Set unit	Selects the default disk unit
yes	no	Interrogate disks	Find which disks are online
yes	no	Interrogate drive	Find disk number of default disk
yes	not needed	Set buffer address	PTDOS: buffer address handled by system (can be user specified when file is opened)
not needed	yes	Indexed seek	Direct seek to block or byte CP/M: can randomly access files at the block level PTDOS: can randomly access indexed files at the byte or block level (block only beyond 65535 bytes)
not needed	yes	Build index	PTDOS: create or update the index of an indexed file
yes	yes	Change file name	
yes	yes	Change file type	
no	yes	Change file attributes	
not needed	yes	Information request	Obtain file type, attributes, blocksize, ID number, etc.
no	yes	Return	Return to Command Interpreter
no	yes	Abort and return	Return with error processing
no	yes	Return/Set trap	Subsequent returns will go to specified address
no	yes	Short reset	Minor abnormal return
no	yes	Error utility	Prints English error messages

Fig. 3. Operating system primitives available with CP/M and PTDOS.

usually use is oriented to the video display of the SOL or VDM board. It allows editing ASCII text files with or without line numbers. ALS-8 format files are also supported. If the file to be edited does not exist, the editor, with your permission, will create the file.

The text displayed on the screen can be scrolled forward or backward a page (16 lines) or a line at a time. The cursor can be moved anywhere on the displayed page, and text modifications can be made at that point. Lines or characters can be added or deleted without retyping any of the text. Blocks of lines can be moved from point to point . . . a very easy text editor to use!

The assembler supplied with the system can assemble files with or without line numbers or files in ALS-8 format. Memory is used only to hold the assembler object code and the symbol table (seven bytes per symbol) of the program being assembled. Thus, large files can be assembled with minimal memory.

A COPY pseudo-instruction can be embedded within the source text to cause the contents of a file to be placed in line in the source text at that point. Source text of tens of thousands of characters can easily be edited and assembled by the COPY pseudo-instruction.

The output listing can be displayed on the video screen, written to a file or suppressed. The symbol table or cross-reference table can be handled in the same way. The object code can be directed to a file or suppressed. There are various output format options for line numbers, page width and page length.

Building a Helios Kit

My Helios was built from a kit and required nine hours to complete. About six hours were required for soldering sockets, components and jumpers on the controller, formatter, power supply and indicator-panel printed-circuit boards. Another three hours were required for mechanical assembly of the disk drive and cabinet. No problems were encountered with the Helios after assembly except for a bad chip on the formatter board.

A disk test program that has several automatic test procedures is provided on cassette. In case the automatic tests indicate any errors, the manual has a long, detailed procedure for testing the many functions of the formatter and controller board. The tests are driven by a test program on the cassette. The test procedure requires a triggered, dual-trace scope. These tests enabled me to discover the bad chip quickly.

The controller board runs very hot. I had to cut holes in the back panel of my SOL and add an extra cooling fan (I added two for good measure); the native cooling of the SOL just wouldn't do it.

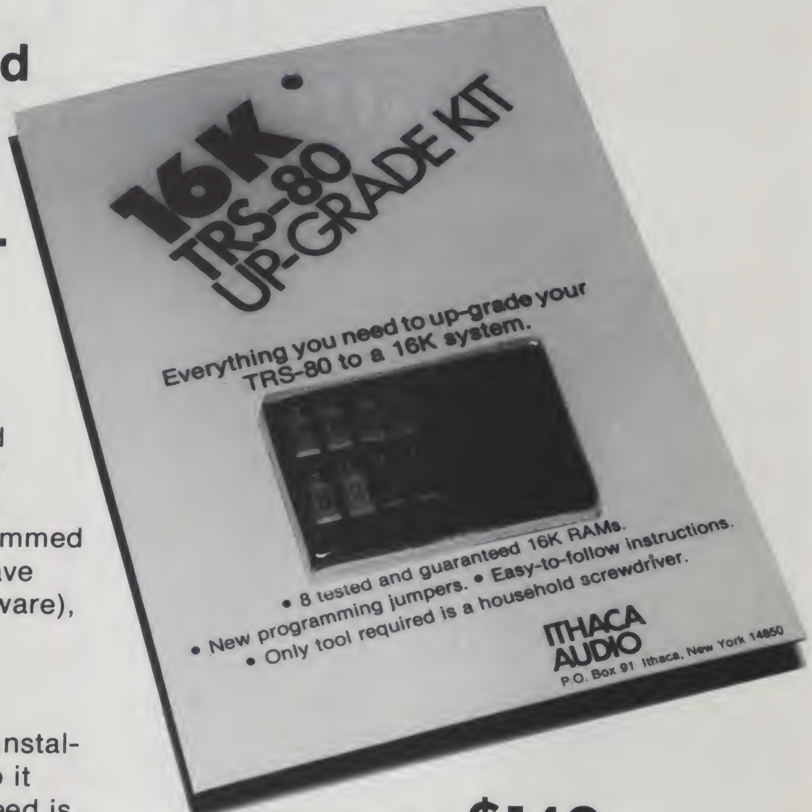
Conclusion

My pocketbook is certainly thinner, but the usefulness and enjoyment of my SOL system has been increased manyfold. My family now uses the cassette recorder without argument. The PerSci drive used is one of the best on the market. The controller and formatter, being on two boards, take up an extra slot in the SOL's limited backplane, but I hope to move the formatter outboard soon. My subjective evaluation is that the system is very good. It was subjected to quite heavy use just before Christmas typing a couple hundred envelopes and three-page letters. It was often operating 16 hours a day.

The PTDOS software is what really separates this system from others of its class. As a software development system, the editor, assembler and DO processor complement each other nicely. Good job, Processor Technology! ■

Introducing the simple TRS-80 Up-grade

Fast, easy, guaranteed expansion to 16K at less than half the price of Radio Shack.



Only $140

Ithaca Audio makes it simple

No false starts and finding you need some little item or special tool. Our Kit contains all the parts: 8 prime dynamic RAMs and a complete set of preprogrammed jumpers. No matter which model you have (even if you later purchase Level II software), you're covered.

Complete Instructions

Our easy-to-follow directions cut installation time to just minutes. You can do it yourself—with no soldering! All you need is a household screwdriver.

100% Guarantee

Like our kit, simple: if a part ever fails, we replace it, FREE.

Available now

Order from your favorite retailer. If by chance he hasn't stocked them yet we'll ship him your Kit right away.

For technical assistance call or write to:

ITHACA AUDIO
Phone: 607/273-3271
P.O. Box 91 Ithaca, New York 14850

Available off-the-shelf at these fine computer dealers.

AL: BIRMINGHAM: Computer Center, (205) 942-8567. HUNTSVILLE: Computerland, (205) 539-1200. **CA:** BERKELEY: Byte Shop, (415) 845-6366. EL CERRITO: Computerland, (415) 233-5010. HAYWARD: Computerland, (415) 538-8080. LOS ALTOS: Computerland, (415) 941-8154. MARINA DEL REY: Base 2, (213) 822-4499. MT. VIEW: Digital Deli, (415) 961-2670. SAN FRANCISCO: Computerland, (415) 536-1592. SAN JOSE: Electronic Systems, (408) 226-4064. SAN RAFAEL: Computer Demo Room Inc., (415) 457-9311. WALNUT CREEK: Computerland, (415) 935-6502. **DE:** NEWARK: Computerland, (303) 738-9656. **FL:** FT. LAUDERDALE: Computer Age, (305) 791-8080. POMPANO BEACH: Computer Age, (305) 496-4999. TAMPA: Micro-computer Systems, (813) 879-4301. **IL:** NILES: Computerland, (312) 967-1714. OAK LAWN: Computerland, (312) 422-8080. PEORIA: Computerland, (309) 688-6252. **KS:** OVERLAND PARK: Personal Computer Center, (913) 649-5942. WICHITA: Computer Systems Design, (316) 265-1120. **KY:** LOUISVILLE: Computerland, (502) 425-8308. **MA:** CAMBRIDGE: Computer Shop, (617) 661-2670. **MD:** ROCKVILLE: Computerland, (301) 948-7676. **MI:** ANN ARBOR: Newman Computer Exchange, (313) 994-3200. ROYAL OAK: Computer Mart, (313) 576-0900. **NJ:** ANDOVER: Atlantic Microsystems, (201) 549-0189. BUDD LAKE: Computer Lab of New Jersey, (201) 691-1984. CLARK: S-100, (201) 382-1318. ISELIN: Computer Mart (201) 283-0600. SUCCASUNNA: Computer Hut, (201) 584-4977. **NY:** BUFFALO: Computerland, (716) 836-6511. ITHACA: Computerland of Ithaca, (607) 277-4888. JOHNSON CITY: Micro World, (607) 798-9800. NEW YORK CITY: Computer Mart of New York, (212) 686-7923. SYRACUSE: Computer Shop of Syracuse Inc., (315) 446-1284. **OH:** CINCINNATI: Digital Design, (513) 561-6733. DAYTON: Computer Solutions, (513) 223-2348. **OK:** OKLAHOMA CITY: Micronics, (405) 942-8152. **PA:** FRAZER: Personal Computer Corp., (215) 647-8463. STATE COLLEGE: Microcomputer Products Inc., (814) 238-7711. **TN:** KNOXVILLE: Eastern Microcomputer, (615) 584-8365. **TX:** AUSTIN: Computerland, (512) 452-5701. DALLAS: KA Electronic Sales, (214) 634-7870. GARLAND: Digital Research Corp., (214) 271-2461. HOUSTON: Houston Computer Mart, (713) 649-4188. **UT:** OREM: Johnson Computer Electronics, (801) 224-5361. **VA:** ALEXANDRIA: Computers Plus, (703) 751-5656. ARLINGTON: Arlington Electronics Wholesalers, (703) 524-2412. **VT:** ESSEX JUNCTION: Computer Mart of Vermont, (802) 879-1683. **CANADA:** ONTARIO: MISSISSAUGA: Arisia Microsystems, (416) 274-6033. TORONTO: Computer Mart Ltd., (416) 484-9708. WINNIPEG: Patrick Computer Systems Inc., (204) 774-1655. **WEST GERMANY:** MUNICH: ABC Computer Shop, Schellingstrasse 33, 8000 Munchen 40 Microcomputer Shop, Toelzerstr, 8, D-815 Holzkirchen. **ISRAEL:** HAIFA: Microcomputer Eng Ltd., Haifa 31-070.

Reader Service—see page 179

Inventory Control with the TRS-80

You may have a large inventory, but you don't need a huge system to handle it.

John A. Yost, Jr.
1041 Hughes Shop Rd.
Westminster MD 21157

In the last few months, I have had several requests for an inventory control program to run on the Radio Shack TRS-80. At first, this seemed too formidable a task for Level I BASIC and only 4K of RAM. However, it is possible if you are willing to work with a few restrictions.

The resulting program allows the user several methods to display his inventory, to update individual items and to save the information on the cassette tape. To accomplish this there are six commands available to the user.

UPDATE: This allows addition, deletion and changes to be made to specified items.

REPORT: Prints all nonzero items on the screen and stops at the end of each screen until ENTER is pressed.

LIST: Requests a range of item numbers to be printed and then prints them using the REPORT routine.

PUT: Writes the item array to the cassette tape.

GET: Reads the item array from the cassette tape.

ITEM NUMBER: Prints a specific item on the screen.

These functions can be simplified or eliminated to make more memory available for the item array.

Overcoming Memory Restrictions

The main problem is how to squeeze many items into limited memory. First, use the item's number as an index into an array. This restricts item number format, since it must be numeric and fall within the range of the array. Now all we have to store are quantity and cost.

The problem here is twofold. There is only one array with one dimension available. Multiple dimensions could have been programmed if it were not for memory restrictions. Each array element requires four bytes of memory. At eight bytes for each item, memory goes fast, so what I have done is combine the quantity and cost into one 6-digit number, which is the number of significant digits we have to work with.

This was accomplished by assigning the quantity to the two high-order digits and the cost to the four low-order digits. This restricts the quantity and cost to integers in the range of 0-99 and 0-9999, respectively. If a dollar-and-cents cost field is required, the cost can be scaled in the pack and unpack routines by either multiplying or dividing by 100. Obviously, this will restrict the cost to a range of 0 to $99.99, and items less than ten cents will display in scientific notation.

Packing the quantity and cost is a simple matter. First the quantity is multiplied by 10,000 to force it into the high-order positions. Then, the cost is added to it. For example, if the input quantity were 15 and the cost $19.95 (input as 1995), the calculation would be as shown in Example 1.

Separating the quantity and cost is more complicated. The quantity is the integer portion of the array item divided by 10,000. Assuming the array item, A(I), is 151995, the unpacking is shown in Example 2.

The program as presented occupies 2407 bytes of memory, which leaves 1176 bytes for array storage on a 4K machine. This means that only 293 items (1176/4-1) can be controlled. On a 16K machine 3365 items can be controlled. Another version of the program allows for about 700 items in 4K of RAM.

Speeding Up

The cassette storage presented a problem at first. When a large number of items were stored, an extensive amount of time (approximately 30 minutes for 400 items) was required for reading and writing the tape. A method of blocking the individual items has been devised; that is, eight items are written with each print. The method may seem a bit awkward, but it works fine[1]. Now, writing 400 items takes slightly less than five minutes. Quite a savings in time and tape.

Special Features

There are several features of the Level I BASIC used that may need further explanation. When executing a line containing an IF statement, the remainder of the line is skipped if the comparison is false; the CLS instruction clears the screen. When used as the object of an IF statement, the * becomes a logical AND operator, and the

[1]TRS-80 User Group Newsletter; Vol. 1, No. 3, p. 6.

```
(Quantity times 10000) plus cost
(15 times 10000) plus 1995
150000 plus 1995
151995
```

Example 1.

+ becomes a logical OR operator.

I used one other feature that leaves me uneasy. This feature allows variables to be used as numeric input. The input variable will be assigned the value of the input variable.

For example, line 110 gives values to several variables. Line 210 asks for you to input a numeric value. If you should input an R or REPORT, variable A will have a value of −1, the value of variable R.

This seems fine at first. However, a problem may occur when an invalid letter is input. The value of that variable will be placed in A and could possibly get you into a routine that you don't want to be in.

If you should get into an unwanted routine, for any reason, all you have to do is hit BREAK and restart the program at line 200 using the RUN 200 command. This feature allows you to start a program at any line just by specifying the appropriate line number.

Line Functions

Lines 100-120. The variables corresponding to the input commands are set to their appropriate values. Then the command list is displayed on the screen.

Lines 200-290. Provides for command input. When a command is entered, it is evaluated. If no match is found, the command list is again displayed on the screen; otherwise, the required routine is performed.

Lines 300-340. Lists the requested item on the screen after decoding the array entry.

Lines 400-490. Update routine. Here an item number, quantity and cost are input. If all inputs are zero, the update routine will end. A range check is then made of the input variables. If any fall outside the range, an error message is displayed and another item is requested. The quantity and cost are then packed together and stored in the array position as indicated by the item number. Another item is then requested.

Lines 500-590. Prints all nonzero items on the screen. Variables B and E hold the beginning and ending values to be used in the report loop. A heading is printed. The array is then scanned for nonzero entries. The array entry is decoded and printed on the screen. After 15 lines are listed, printing is stopped until ENTER is pressed. The paging routine does this work (lines 9300-9340).

Lines 600-630. Prints a particular range of items. The first and last item numbers are requested. Control is then transferred to the report routine (lines 500-590) to do the actual printing.

Lines 700-750. Writes the item array to tape.

Lines 800-850. Reads the item array from tape.

Lines 9000-9090. Prints the commands on the screen.

Lines 9100-9110. Decodes the array item and computes total cost.

Lines 9300-9340. Paging routine. After ENTER is pressed, indicating that you are ready for the next page, the screen is cleared and a new heading is printed.

This program is presented not as a practical solution to inventory control, but to demonstrate what can be done on a minimal system. Some of the techniques used may be helpful in other programs. ■

```
Quantity = integer portion of A(I) divided by 10000
Cost = A(I) minus quantity times 10000
Quantity = INT(151995/10000) = 15
Cost = 151995 – 15•10000 = 1995
```

Example 2.

```
0100 REM  *****   INITIALIZE  *****
0110 N = 0:U = 0:R = -1:L = -2:P = -3:G = -4
0120 GOS.9000
0200 REM  *****   MAIN CONTROL   *****
0210 IN."COMMAND";A
0215 IF A>N T.P."ITEM # OUT OF RANGE":G.200
0220 IF A>0 T.GOS.300:G.200
0230 IF A = U T.GOS.400:G.200
0240 IF A = R T.GOS.500:G.200
0250 IF A = L T.GOS.600:G.200
0260 IF A = P T.GOS.700:G.200
0270 IF A = G T.GOS.800:G.200
0280 GOS.9000
0290 G.200
0300 REM  *****   PRINT REQUESTED ITEM   *****
0310 I = A:GOS.9100
0320 P."ITEM # ";A;"   QTY ON HAND = ";Q;"   COST = ";C;
0330 P."   TOTAL = ";T
0340 RET.
0400 REM  *****   UPDATE ROUTINE   *****
0410 P."ENTER ALL ZEROS TO END UPDATE"
0420 IN."ENTER ITEM #, QTY,COST";I,Q,C
0430 IF (I = 0)*(Q = 0)*(C = 0) T.G.490
0440 IF (I<1) T.P."INVALID ITEM #":G.420
0450 IF (Q<0) + (Q>99) T.P."INVALID QTY":G.420
0460 (C<0) + (C>9999) T.P."INVALID COST":G.420
0465 IF I>NT.F.B = N + 1TOI:A(B) = 0:N.B:N = I
0470 A(I) = (Q*10000) + C
0480 G.420
0490 RET.
0500 REM  *****   REPORT ROUTINE   *****
0510 B = 1:E = N
0520 GOS.9320
0530 F. I = BTOE
0540   IF A(I) = 0 T.580
0550   IF X>14 T.GOS.9300
0560   GOS.9100:X = X + 1
0570   P.TAB(5);I;TAB(15);Q;TAB(23);C;TAB(32);T
0580 N.I
0590 RET.
0600 REM  *****   LIST ROUTINE   *****
0610 IN."ENTER RANGE TO BE PRINTED";B,E
0615 IF (B<1) + (B>N)T.P."INVALID BEGIN NUMBER":G.610
0616 IF (E<B) + (E>N)T.P."INVALID END NUMBER":G.610
0620 GOS.520
0630 RET.
0700 REM  *****   WRITE TAPE ROUTINE   *****
0710 CLS:P."LOAD TAPE TO BE WRITTEN TO"
0720 IN."PRESS ENTER WHEN READY";A$:P."WRITING TAPE"
0725 E = INT(N/8)*8:P.#N
0730 F.I = 1TOE STEP 8
0732 D = A(I):F = A(I + 1):H = A(I + 2):K = A(I + 3):M = A(I + 4):Z = A(I + 5):O = A(I + 6)
0733 S = A(I + 7)
0735 P.#D;",";F;",";H;",";K;",";M;",";Z;",";O;",";S
0737 N.I:F.I = E + 1TON:P.#A(I):N.I
0740 P."DATA SAVED"
0750 RET.
0800 REM  *****   READ TAPE ROUTINE   *****
0810 CLS:P.:P."LOAD TAPE TO BE READ"
0820 IN."PRESS ENTER WHEN READY";A$
0825 IN.#N:E = INT(N/8)*8
0830 F.I = 1TOE STEP 8
0832 IN.#D,F,H,K,M,Z,O,S
0833 A(I) = D:A(I + 1) = F:A(I + 2) = H:A(I + 3) = K:A(I + 4) = M:A(I + 5) = Z:A(I + 6) = O
0835 A(I + 7) = S
0837 N.I:F.I = E + 1TON:IN.#A(I):N.I
0840 P."DATA LOADED"
0850 RET.
9000 REM  *****   PRINT COMMANDS ROUTINE   *****
9010 CLS
9020 P." * * * * *  C O M M A N D S  * * * * *"
9030 P."'UPDATE' OR 'U': TO UPDATE ITEMS"
9040 P."'REPORT' OR 'R': TO PRINT ALL ITEMS"
9050 P."'LIST' OR 'L': TO PRINT A RANGE OF ITEMS"
9060 P."'PUT' OR 'P': TO WRITE ITEMS TO TAPE"
9070 P."'GET' OR 'G': TO READ ITEMS FROM TAPE"
9080 P."ITEM NUMBER   : TO LIST A PARTICULAR ITEM"
9090 P.:P.:RET.
9100 REM  *****   DECODE QTY AND COST ROUTINE   *****
9110 Q = INT(A(I)/10000):C = A(I) - (Q*10000):T = Q*C:RET.
9300 REM  *****   PAGING ROUTINE   *****
9310 IN."PRESS ENTER FOR NEXT PAGE";A$
9320 CLS
9330 P."   ITEM #";TAB(15);"QTY";TAB(23);"COST";TAB(32);"TOTAL"
9340 X = 1:RET.
```

Program listing.

Onward with the COSMAC Elf!

Ready to expand your minimum memory Elf into a full-blown 64K system? Whether it's all the way to 64K, or just another 256 bytes, you'll find the following guidelines invaluable.

Jeff Duntemann
6208 N. Campbell Ave.
Chicago IL 60659

A recent tally showed that RCA's COSMAC microprocessor was the fifth most popular chip among hobbyists. Considering the tiny handful of commercial COSMAC computers available, it's a good bet that most COSMAC machines in hobbyists' hands are handbuilt, from scratch, following the COSMAC Elf series in *Popular Electronics* ("Build the COSMAC Elf," by Joseph Weisbecker, August, September 1976; March, July 1977).

The Elf used only one page of RAM. Although the writer will attest that this is plenty of room for a beginner to get lost in, Elf owners soon get an itch to expand their memory systems. Going from a single 256-byte page of RAM to many is not simply a matter of plugging in the chips. If you're considering adding memory to your Elf, read on—this article may save you some headaches.

Memory Address Multiplexing

There's a reason the original Elf had only one 256-byte page of memory. The CDP 1802 IC has only eight memory address pins. Since $2^8 = 256$, the chip has just enough pins to address 256 bytes of memory directly. A simple system like the Elf can accept this and do very well without any additional fooling around. Programs written for 256-byte systems are simpler, too.

Addressing more memory requires *memory address multiplexing*. This is jargon for a neat trick RCA uses to put a 16-bit address on eight memory address pins. First, the 16 bits, called A0 through A15, are split into two 8-bit groups. A8 through A15 are called the higher-order bits, and they appear on the memory address pins first. The Elf does not use these in its unexpanded form.

While these bits are present on the pins, a timing pulse called TPA appears on its own pin. If you wish to use the higher-order bits, you must set them in a latch (like the CD4042) external to the 1802 IC. The TPA pulse strobes the latch and saves the higher-order bits. These bits then vanish from the memory address pins, to be replaced immediately by A0 through A7, called the lower-order address bits. These are the eight bits used by all COSMAC computers to select individual bytes within a 256-byte page of memory.

The higher-order bits are used to select an individual memory page out of several pages in a system. Not all of these bits are used in most multipage systems. Only the first higher-order bit (A8) is needed to select one page out of a two-page system. The first two higher-order bits will select one page in a four-page system, and so on. Using all eight higher-order bits would enable you to select a single page out of 256 pages of memory. That would be a COSMAC "full gallon," 65,536 bytes of memory, and probably a little more than you could afford this payday.

To review: In COSMAC expanded memory systems, the higher-order bits (as many as necessary) are used to select one page out of many pages of memory. The lower-order bits (all eight) are used to select a single byte within a 256-byte page.

Expansion Hardware Kinks

Expanding memory using 2101-type RAM is not difficult. In fact, it's a good deal easier than using 2102-type RAM. Each memory page consists of two 2101 RAM chips. To expand memory, wire up as many pairs of 2101s as you desire in parallel with the original page. MOS inputs are open circuits, and may be freely wired in parallel ("wired-OR," in the jargon). The output pins of the 2101 are Tristate, and are driven to an open-circuit state when not needed. The various timing pulses from the 1802 ensure that only one set of outputs is turned on at any one time, so it is alright to wire outputs in parallel as well.

The only 2101 pin that must be treated specially is pin 19. This is the chip-select pin. A low on this pin selects *the chip*. A high "floats" all the outputs in a high-impedance state and ignores all commands on the control pins. It places the chip in a sort of electronic limbo in which it is essentially cut out of the memory system. The idea behind an expanded memory system is to keep all 2101 chips in limbo except for the two that make up the selected memory page.

Connect the two pin 19s within each memory page together, but keep each page's pin 19s separate. This lead from each page will be the page-select lead. There are two good page-select logic methods for expanded COSMAC systems. I've sketched them out in Figs. 1 and 2.

For a two-page system, one-

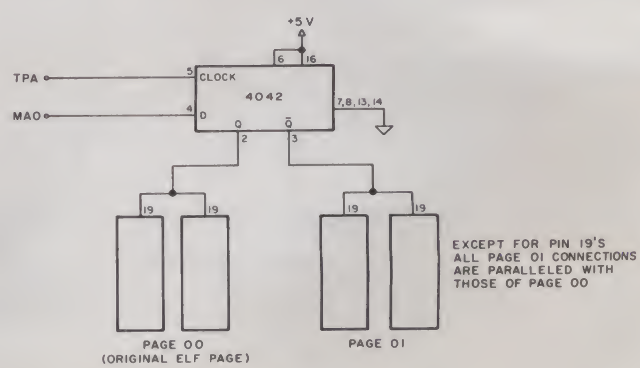


Fig. 1. Two-page system.

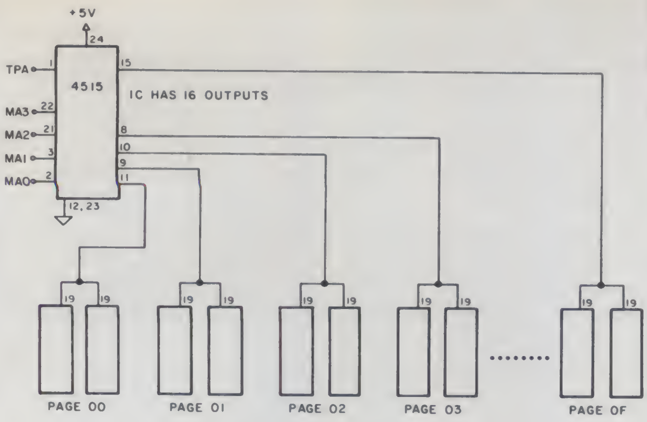


Fig. 2. Multipage system.

fourth of a CD4042 quad latch IC will do all the latching and decoding you will need. Each of the 4042's latch sections has two outputs. One output repeats the input, and the other inverts the input. Connected as in Fig. 1, the Q and $\overline{Q}$ outputs will always be different, assuring that only one memory page of the two will be selected at any one time. Make sure that the other three D inputs to the 4042 are grounded. The unused outputs may be left open, as is the rule with all CMOS ICs.

For a larger memory system with up to 16 pages of memory, use the configuration shown in Fig. 2. The CMOS CD4515 IC contains both a four-bit latch and a four-to-16 line decoder. At every TPA pulse, A8-A11, the first four higher-order memory address bits, are saved in the latch and decoded to select one page out of the system. Only one of the 16 output pins of the 4515 will be low at any given time. The others will be held high. One page-select lead goes to each 4515 output, starting with output 0. The operation should be pretty obvious.

This second system is not a 16-or-nothing choice. You may use it with as little as two pages of memory and add pages at any time, up to the maximum 16. The unused outputs may be left open.

If you decide to expand to more than about eight pages, it's a good idea to buffer your memory address lines. This involves following each of the 1802's memory address outputs with one section of a CD4050 hex driver, and using the driver outputs as you would the memory address outputs. Each driver output can supply a good deal more current than the individual outputs from the 1802 IC itself.

You might question how a CMOS input, which is actually an open circuit, can require an input current. Each CMOS or MOS (in this case) input is actually a very tiny capacitor. When a capacitor goes from an uncharged to a charged state, it draws a very brief surge of current. When a MOS input goes from a logic low to a logic high, it charges and draws a tiny current pulse. For one or a handful of MOS inputs this current is insignificant, but for a large number of inputs it adds up. If the current becomes large enough, the driving output is loaded too heavily and slows down.

In the case of memory address lines, this can cause program execution problems and possibly overheat the 1802. CD4050 drivers have the "guts" to charge a large number of MOS inputs quickly, while presenting only one CMOS input to the 1802's memory address outputs. Loading down your memory address lines is not a good idea if you expect your system to run at any respectable speed. COSMAC is the fastest non-bipolar processor around, passing even the great Z-80 by two whole megahertz . . .why waste it?

Buffering the address lines is also essential if you ever want to add a memory address display to your computer. The same buffers will drive both the address display and all the memory you want to add. For the price of two 4050s it's a worthwhile addition to your computer.

While you're adding memory to your computer, you might want to consider using CMOS RAM instead of current-hungry 2101s. You have your choice of RCA's CDP 1822 and the 5101, which is offered by several manufacturers and very well described in Don Lancaster's *CMOS Cookbook*. Both these chips are pin-for-pin compatible with the 2101, and can be plugged into the same sockets you wired up for 2101 RAM. The reduction of power consumption using CMOS RAM is shocking: on the order of 1/10 to 1/40, depending on processor clock speed.

One final advantage is that CMOS RAM can be made "nonvolatile" (able to hold programs when system power is off) by adding a switch, some diodes and a small battery. I will only add that any 2101 pin-compatible CMOS RAM IC will work as well as the CDP1822 in that circuit. The 5101 is much less expensive and a good deal more widely available.

A COSMAC Memory Address Display

I'm sure you've had those hair-pulling times when you load a laboriously long program in through the toggle switches, flip RUN up and . . . nothing. Has the program stopped in its tracks? . . . gotten caught in an endless loop? You have no way of knowing which instructions the computer is executing—if indeed it is executing any at all. Those are the times when a memory address display really earns its keep.

The circuit in Fig. 3 will display the address of the byte the computer is currently fetching, anywhere within the 65K memory address range of the CDP1802. It also keeps track of your "place" while you're toggling in a long program. If you've ever lost count while toggling in 12 NOP instructions in a row you will appreciate that feature.

The Texas Instruments TIL311 display device contains a latch, decoder, constant-current LED driver and LED dot-matrix display. It creates more natural and pleasing hexadecimal digits than seven-segment displays, especially for the alphabetic hex characters. It *is* expensive, but if you shop around you can usually find them for less than $8 apiece.

The circuit given in Fig. 3 covers all 65K of memory that the 1802 can address. For a single-page system the last two digits will suffice. Unless you have more than 16 pages of memory you need only the last three. In any case, to use the

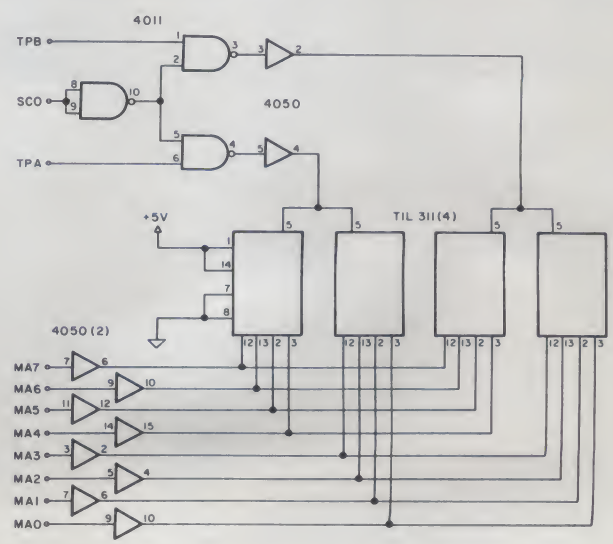


Fig. 3. COSMAC memory address display.

TIL311 you must buffer your memory address lines. Also bear in mind that each TIL311 draws more than 100 milliamps from your power supply.

Using a memory address display is not just as simple as looking at it. Your computer executes close to 100,000 instructions per second; the digits go by rather quickly. If you start your computer running and the displays read some number "cleanly," that is, without any extra LED dots lit, the program has stopped, and the displayed digit gives the address of the last instruction fetched.

In most cases you will see some strange pattern of dots that looks like several numbers overlaid upon one another. That is exactly what you see. The computer is looping, and the displays are showing the addresses of all instructions within the loop, so quickly that the eye sees them as one. In some cases you may be able to puzzle out some of the numbers for very short loops: "1" atop "2," for example.

```
MA

00   90 B1 B2 A3        Set high-order bytes & RA.0 to 00
04   F8 1E A1           Point R1.0 to synthetic op code location
07   F8 FF A2           Put FF in R2.0
0A   E2                 X=2
0B   7B                 Turn Q on
0C   3F 0C              Wait for Input pressed
0E   37 0E              Wait for Input released
10   83                 Put R3.0 in D
11   39 19              If Q=0 go to M(19)
13   F9 80              D OR 80 = D
15   7A                 Turn Q off
16   51                 Store D at M(R(1))
17   30 1E              Go to M(1E)
19   F9 90              D OR 90 = D
1B   51                 Store D at M(R(1))
1C   7B                 Turn Q on
1D   13                 Increment R3
1E   XX                 Put Register X.X in D (synth. op code)
1F   52                 Store D at M(R(2))
20   64 22              Output register contents to hex display
22   30 0C              Go to M(0C)
```

Program 1.

```
MA

00   90 B2 B3 B5 B6     Initialize high-order registers
05   F8 FE A2           Set stack address in R2.0
08   F8 12 A3           Set MAIN program counter in R3.0
0B   F8 31 A5           Set BSUB program counter in R5.0
0E   F8 41 A6           Set HSUB program counter in R6.0
11   D3                 P=3.  Start MAIN
12   D5 B1              P=5.  Call BSUB & put byte in R1.1
14   D5 A1              P=5.  Call BSUB & put byte in R1.0
16   6C                 Read toggles; put byte in D
17   3A 22              Go to M(22) if D≠0
19   F8 1D A0           Re-point R0.0 at M(1D)
1C   D0                 P=0
1D   91 B3 81 A3        Put contents of R1 into R3
21   D3                 P=3.  Jump to keyboard-selected address
22   F6                 Shift D 1 bit right
23   3B 2A              Go to M(2A) if DF=0
25   D5                 P=5.  Call BSUB & put byte in D
26   E1 64              X=1.  Display M(R(1)) on hex readouts
28   30 25              Loop and display next memory location
2A   D5                 P=5.  Call BSUB & put byte in D
2B   E1                 X=1
2C   51                 Store D at M(R(1))
2D   64                 Display M(R(1)) on hex readouts
2E   30 2A              Loop and store another byte at next location
30   D3                 P=3.  (Return from BSUB)
                           BSUB
31   D6                 P=6.  Call HSUB & put nibble in D
32   FE FE FE FE        Shift D four bits left
36   A0                 Store D temporarily in R0.0
37   D6                 P=6.  Call HSUB & put nibble in D
38   80 F1              Combine nibbles via OR function
3A   52                 Store D at M(R(2))
3B   64 22              Display byte and decrement R2
3D   30 30              Return
3F   02                 Put M(R(2)) into D
40   D5                 P=5.  (Return from HSUB)
                           HSUB
41   E2                 X=2
42   FC 01              Add 01 to D
44   FA 0F              D AND 0F = D
46   52                 Store D at M(R(2))
47   62 22              Output D to encoder chip & decrement R2
49   3D 41              Loop again if no key is pressed
4B   7B                 Turn Q on
4C   F8 09 B4           Load 09 into R4.1
4F   24                 Decrement R4
50   94                 Load R4.1 into D
51   3A 4F              Loop again until R4.1=0
53   7A                 Turn Q off
54   35 54              Wait until key is released
56   30 3F              Go to M(3F)
```

Program 2 (EHOPS-65K).

For longer loops, rapidly flip RUN up and down. Each time you stop the program, the displays will randomly read one address within the loop. Do this often enough, and all addresses in the loop should show up at least once. If you keep track of the number of times a particular address comes up, you might be able to track loops within the loops by spotting where in the program the computer seems to be spending an inordinate amount of time.

With a little practice, you'll find that an address display is the most cost-effective debugging device you can add to your computer.

Beware the Rest of That Register!

People who expand the Elf's memory system beyond its single page soon discover that programs written with only a single page in mind work strangely, if at all, on multipage systems. Like most everything else about the COSMAC architecture, it all comes down to the general-purpose registers, and to the way they address memory.

In the basic single-page Elf, program counters, data pointers, stack pointers or *any* stored memory address need only be eight bits long and may be stored in the lower half of a general-purpose register. The upper half may contain nothing, or anything at all, and still be ignored because the upper half of a register contains the higher-order bits of a memory address. These function only to select the proper memory page in a multipage system, and in the Elf the single page is selected all the time.

As soon as you add even one additional page of memory you must watch what is in the upper halves of your registers. If you don't, your program counter may suddenly start trying to execute instructions in a memory page that doesn't even exist! Your pointers can easily end up pointing to the Twilight Zone.

Even short programs in multipage systems need to be modified. Many COSMAC users operate under the severely mistaken notion that all general-purpose registers reset to 0000 when power is applied. *Not true!* Only R0 has any power-on reset provision. The other 15 registers, after power-up, may contain any sort of garbage at all and usually do.

To prove it, turn your power off and on again, and load Program 1. After flipping RUN up, nothing will happen until you press INPUT. Then the data display will show the contents of R0.0. Push INPUT again, and the display will show R0.1. The Q light will come on, indicating that the display is of a higher-order register byte. A third press of INPUT will show R1.0, and so on.

You can step through the 1802's entire complement of registers by successively pressing INPUT. Each time the Q lamp lights, the display is a higher-order byte. You will see that the registers contain a great deal of trash after power-up.

For short programs, the easy way to preset your higher-order register bytes is by making use of R0's power-on reset function. As the first order of business in a program, bring R0.1 into D and copy it into as many higher-order registers as you intend to use to hold any sort of memory address:

90 B1 B2 B3 B4 B5

etc. This will put 00 into your higher-order register halves and prevent pointer problems. Of course, if you're using an operating system and locating programs in some other memory page, set the higher-order halves of your pointer registers to the proper page:

F8 01 B1 B2 B3 B4 B5

for a program running in page 01. Note that this method uses immediate data and cannot use R0's power-on reset.

The simplest way to keep all

this straight is to train yourself to think in terms of 16-bit memory addresses rather than the Elf's 8-bit addresses. Don't think of the two halves of your general-purpose registers as separate and independent entities. For static-byte storage, sure, but *it takes both halves of a register to hold an address*. Your data pointers may no longer look like 08 or FF. They must resemble 03C0 or 0199 or 00FF. Once you break yourself of the 8-bit Elf habit, the whole thing will begin to seem natural and inevitable.

Perhaps the hardest part of expanding your memory system will be rewriting all your software with expanded data pointers and program counters. To save you a little work (and to provide an example of how to go about it), I've rewritten the EHOPS operating system to run with expanded memory. Call it EHOPS-65K. It can handle any amount of memory COSMAC can address.

Take a close look at Program 2. Notice that the higher-order register halves are immediately set to 00. Also notice that it calls BSUB *twice* for each starting address, storing the starting address in *both* halves of R1 (MA 0012-0014).

Thus, when you flip RUN up you must tap two bytes into the hex keypad, regardless of what function you select with the toggle switches. The first byte you tap in will be the higher-order byte of the starting address, followed by the lower-order byte. After the starting address is entered, you may use EHOPS-65K exactly the same as you would use EHOPS-256.

EHOPS-256 uses a trick that doesn't work in expanded memory systems, and it merits a closer look as an example of how not to do things. In order to run a program that has been located elsewhere in memory, EHOPS-256 simply asked for a one-byte starting address, stored it in R1.0, and then made the jump to the program by copying R1.0 into the current program counter, R3.0. Recall that the number in a program counter is the address of where the computer is "at" in executing instructions.

In an expanded memory system, you cannot completely change a two-byte program counter with only one instruction. You must change it a byte at a time, and as soon as you change either half of your current program counter, your program instantly vanishes to begin running in some other area of memory, leaving behind the other half of the change routine like an abandoned child.

EHOPS-65K does not alter the current program counter. To run a program in another area of memory, it temporarily drops out of program counter R3 into R0, quickly copies the contents of R1 (the starting address) into both halves of R3, and then assumes R3 as program counter once again. This immediately sends program execution to the starting address tapped in on the keypad.

Note that in order to accomplish this trick R0 must be specially repointed to "pick up" the program precisely where the program "abandons" R3. All in all, the method is awkward and uses a lot of op code to execute a simple jump.

To sum up, *don't tamper with the current program counter*. As soon as your system grows beyond a single page of memory, such tricks instantly become disasters.

Now, as an exercise, rewrite ETOPS-256 to run in an expanded memory system.

And Don't Overlook the Obvious

You might also wish to avoid a mistake made by the writer during an ambitious expansion of his COSMAC system. He industriously added nine pages of 2101 RAM, plus a four-digit memory address display using TIL311s. The moment he hit the "on" switch, the power supply regulator vanished in a puff of smoke. Use a thermally protected regulator, or at least a fuse, and coping with COSMAC expansion won't be that hard at all.■

Duane Amundson
1214 Aloha Lane
Clearwater FL 33515

Build a $50 TVT!

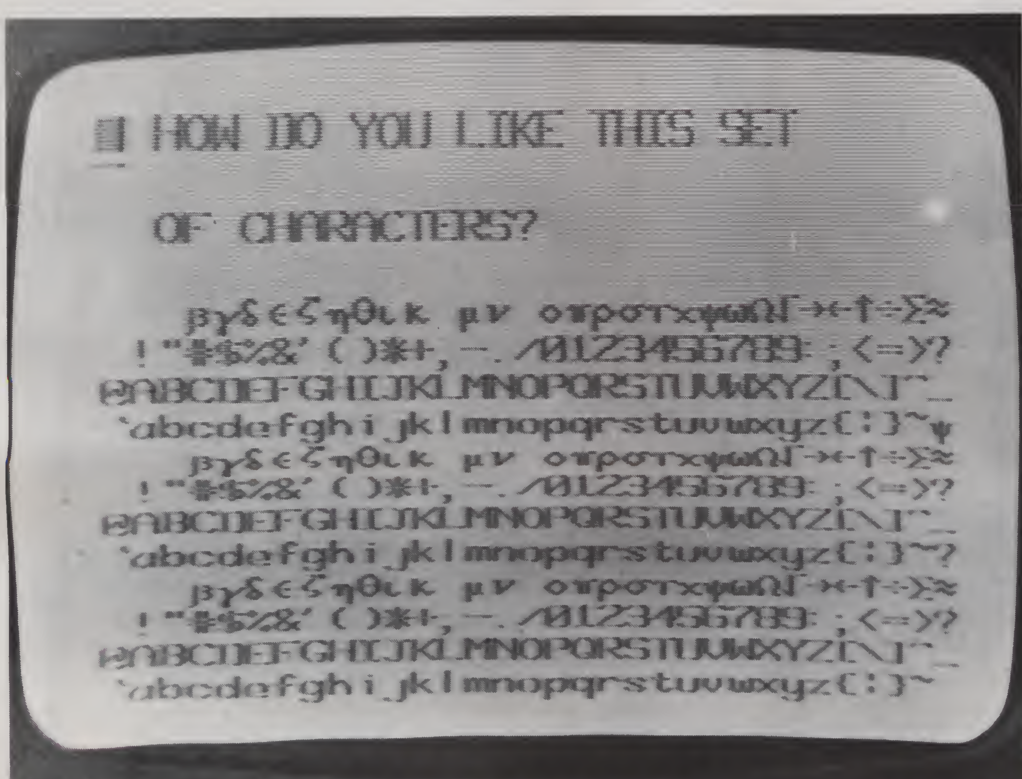
Most of us want to have video displays for our microcomputers. The following design puts everything that you need on one small board.

Many microprocessor-based systems produce data that is best displayed on a CRT. We are going to discuss a circuit that generates a composite video signal to drive a video monitor or TV set and can be built for about $50!

Duane Amundson is senior engineering aide at the ECI Division of E-Systems, St. Petersburg FL.

The signal, when injected into the video amplifier of a TV set, displays a page of 16 32-character lines for a total of 512 characters. Its memory has a capacity of 1024 characters, or two pages. Either page can be selected for display. The characters can be entered either directly from a keyboard or microprocessor that produces ASCII characters. The circuit includes a cursor generator to simplify editing the displayed data.

It also includes an MCM6571A character generator, which produces a complete upper and lowercase English and lowercase Greek character set plus many special symbols. The MCM6571A is directly interchangeable with most of the 6570 series, thus allowing other character sets, such as German or Japanese, to be used. The memory output is made available to allow a microprocessor to read the data displayed on the screen. The vertical and horizontal sync pulses are brought out to allow greater flexibility in connecting the circuit to a TV set. If built from standard TTL ICs, the circuit requires a total input power of about 5 W. If low-power TTL is used, the power requirement can be reduced to about 3 W.



Theory of Operation

As shown in Fig. 1, the circuit can be logically divided into seven sections: horizontal, vertical, write, selector, cursor, memory and video.

The horizontal section contains the master clock. This is divided to produce horizontal-sync and blanking signals. The vertical section further divides the output of the horizontal section to produce the vertical-sync and blanking signals. The combined horizontal and vertical sections produce a 9-bit address, which defines the location in the memory of the character position being scanned by the CRT. The *write* section controls the writing of the new characters into the memory. The selector section determines which address is

sent to the memory, the read address produced by the horizontal and vertical sections, or the write address produced by the write section. The cursor section produces a signal that appears on the CRT as a line under the character position into which the write section will direct the next new character to be written. The memory section stores the characters that have been entered and produces the video output signal required to display these characters on a CRT. The video section combines the video signal from the memory, the cursor signal from the cursor section and the horizontal and vertical sync and blanking to produce the composite video output signal that drives the video monitor or TV set.

Construction

The circuit can be built with a printed circuit board, wire-wrap sockets or point-to-point wiring. I used a printed circuit board (see Fig. 2a) because it results in a much neater and easier way to repair the final product, as can be seen in the photo of the front and back of the PCB. The components on the PCB are identified in Fig. 2b. The parts list gives a more complete description of each part. If a little discretion is used in purchasing parts, the circuit can be constructed for a total cost of about $50. If a PCB is used, it should be very carefully inspected before any parts are installed. If any printed wires are broken or shorting to adjacent wires, it is easier to repair them at this time. To keep the board size at a conveniently small 6 X 8.7 inches, the wires and parts must be close together. Each

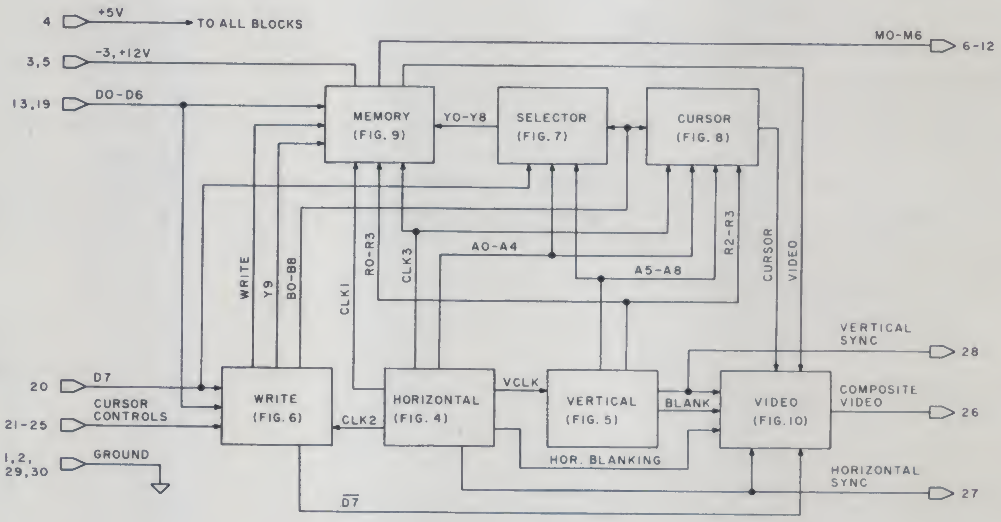


Fig. 1. Overall block diagram.

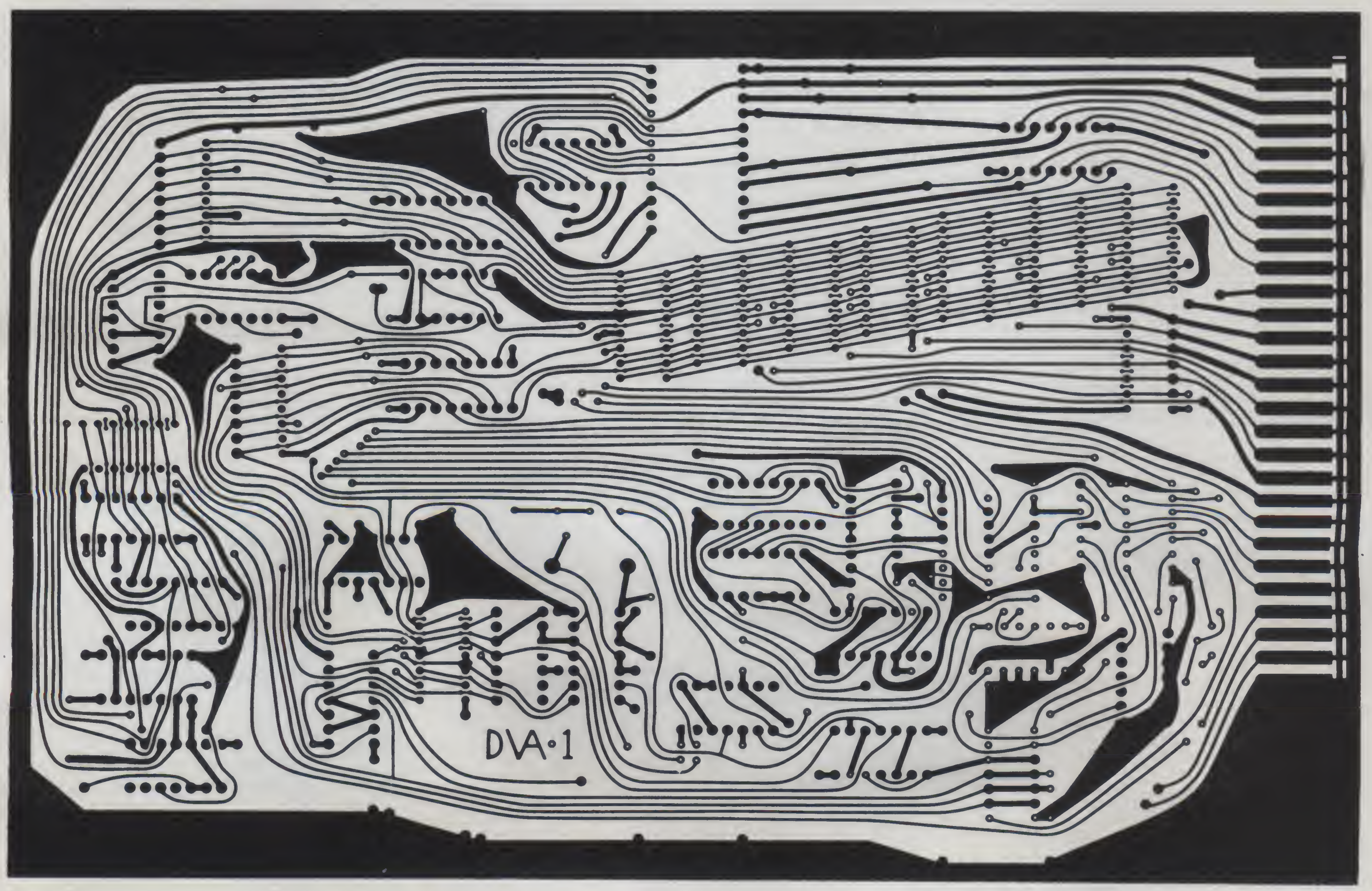


Fig. 2a. PC board artwork. (Fig. is 80 percent of artwork's full size.)

solder connection should be inspected as it is made to see that no solder bridges are formed.

The holes for the jumper wires and all other components except the crystal should be drilled with a #70 drill bit. The crystal pins require a #55 drill bit, and #22 or #24 wire can be used for the jumpers, using insulating tubing where necessary to prevent their touching each other. The photo shows where to use insulating sleeving. Since the 2102s and the 6571A are MOS devices and may be damaged by a static charge, it is safer to install these last, taking reasonable care to prevent static buildup. Unless the 7400 series ICs have all been tested or are known to be good, I recommend checking each one with an ohmmeter from Vcc to ground. On most of the TTL ICs, this is from pins 14 to 7; however, 7473s are pins 4 to 11, 7493s are pins 5 to 10, and the remaining 74193s, 7485s, 74157s and 74165 are pins 16 to 8. It is much easier to locate a shorted IC at this time than after it has been soldered into the PCB with 39 other ICs.

After all the parts have been carefully soldered in place and the finished board has been inspected for bad solder connections, it is ready to use. Interface the board as shown in Fig. 3. The board shown is designed for a 30-pin edge connector with .156 inch pin spacing. This allows for two ground pins at each end of the connector. The D0-D6 lines bring the 7-bit ASCII code from the keyboard to the TV readout board. The D7 signal is a positive strobe pulse generated by the keyboard to indicate that a key has been pressed and the ASCII code has been set up on the D0-D6 lines and is ready to be read. The cursor control lines can each be connected to a key on the keyboard or a push-button switch, but must be held low when the key is not pressed and high when the key is pressed. The high state could be a short pulse.

The composite video is best connected to the TV video amplifier circuit with a 93 Ohm coax line terminated with a 100 Ohm resistor, although a twisted pair of wires could be used. When first turned on, the board should fill the TV screen with a random set of characters. Many keyboards have a rubout key that produces a home command and writes a solid rectangle in the upper left corner of the screen. If the circuit does not do everything it should, the description of each part of the circuit board in the circuit analysis will help locate the problem.

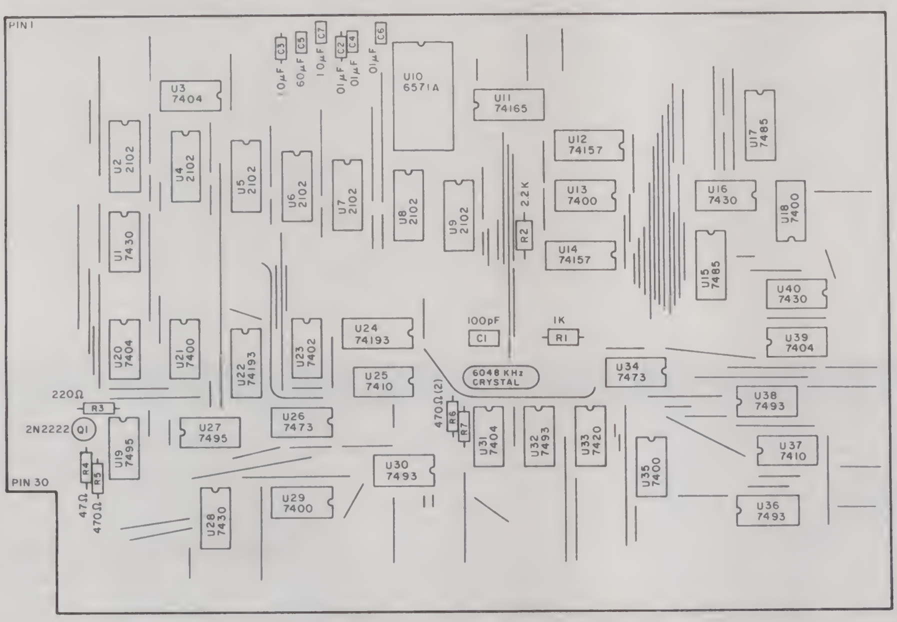


Fig. 2b. Parts location and identification.

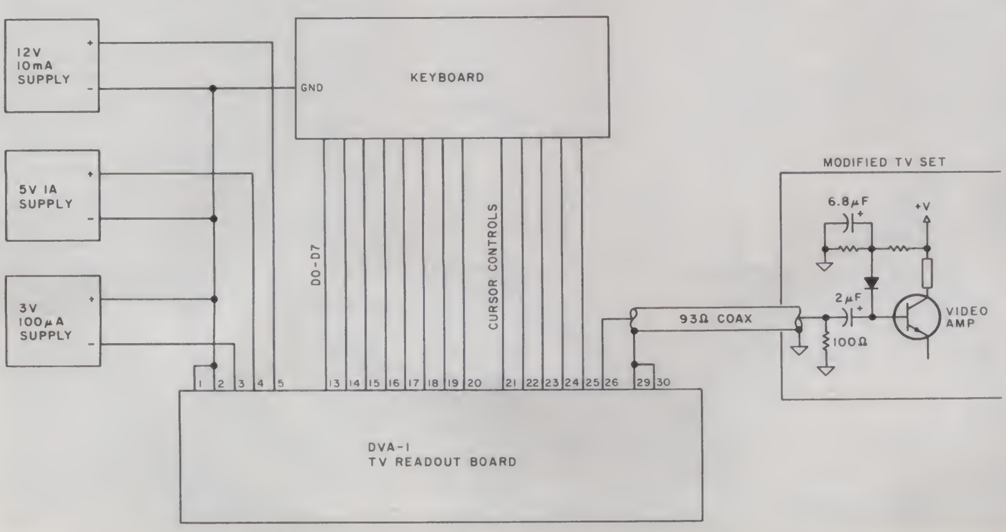


Fig. 3. Power supply, keyboard and TV set connections.

Circuit Analysis

With an oscilloscope, it is possible to isolate any trouble in the circuit. The numbered blocks in Fig. 1 and Figs. 4 through 10 indicate the pin numbers of the PCB edge connecter. The character generator requires −3 V and +12 V in addition to +5 V. All other ICs require only the +5 V supply. Except for U11 pin 1 and Q1, all signals on the board are either a logic low or a logic high, or alternating between these two states. A logic low is any voltage less than .8 V. A logic high is any voltage between 2.0 and 5.0 V. If an oscilloscope is not available, a 2k VOM meter can provide useful troubleshooting information. It can distinguish between a low, high or alternating condition. However, it may not distinguish between an alternating condition and a faulty IC.

The horizontal section is shown in Fig. 4. CLK1 is a 6.048 MHz square wave, which is used by the character-bit shift register. CLK2 is a 1.512 MHz square wave, which is used by the write/count shift register. CLK3 is a 756 kHz square wave used by the character-bit shift

✓ Reader Service—see page 179

register and the cursor-delay shift register. A0 through A4 is the horizontal part of the read address. This is sent to the selector and the cursor generator. A0-A4 are all square-wave signals: A0 = 378 kHz; A1 = 189 kHz; A2 = 94.5 kHz; A3 = 47250 Hz; A4 = 23625 Hz.

The 6048 kHz oscillator consisting of U31ab, R6, R7 and the crystal is the master clock, which provides a stable reference frequency for the entire board. U30, U34b and U32 are the horizontal part of the frequency dividing chain. U35a and U31f generate a reset signal at pulse 384. Therefore, QD, which is pin 11 of U32, which is the output of the last stage of the horizontal divide chain, has a frequency of 15750 Hz (6048000 ÷ 384). There are eight CLK1 cycles for each change of state of A0, the least significant bit of the address. This means that each character position is eight bits, or 1.32 μs, wide. Therefore, each horizontal scan requires 48 (384 ÷ 8) character positions. These are defined by A0-A4 plus QD.

Because only positions 0-31 are displayed and 32-47 are blanked, QD is not required as part of the address. At the start of each character, the character code is loaded from the memory into the shift register, U11, at the same time that the address is changed. Because a short time must elapse before the output of the memory changes, the character displayed in a character position will be the one stored at the immediately preceding address. Therefore, the blanking circuit, which consists of U31e and U29, must blank at 33 and unblank at 1 in order to display addresses 0-31.

U31d, U33 and U35cd generate the horizontal sync pulse and VCLK (which is simply an inverted sync pulse). The sync pulse starts at 37 and ends at 43, making it six characters, or 7.9 μs, wide. This makes the front porch four characters, or 5.3 μs, and the back porch six characters, or 7.9 μs, wide. The width of the blanking signal, 16 characters, is 21.2 μs.

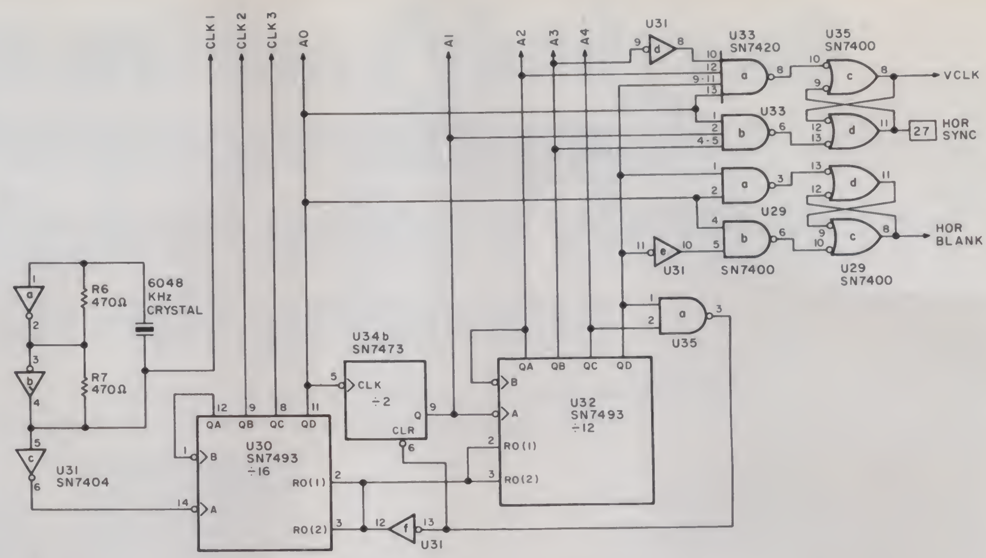


Fig. 4. Logic diagram of horizontal circuits.

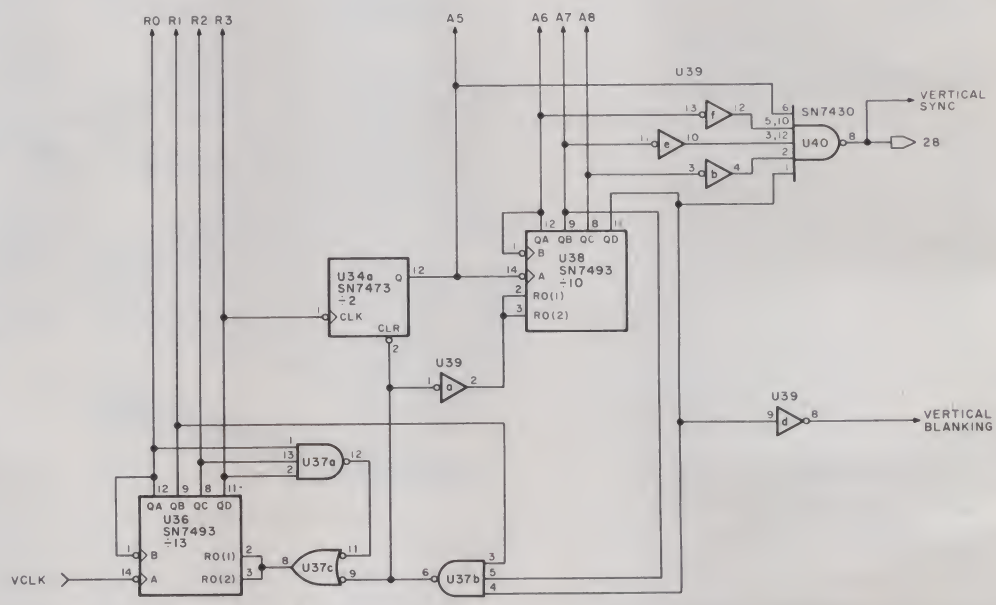


Fig. 5. Logic diagram of vertical circuits.

The vertical section is shown in Fig. 5.

R0, R1, R2, R3 are the numbers of the horizontal row of the character being displayed, and A5, A6, A7, A8 are the line numbers. This section requires VCLK as the input to the first counter. U36 and U37ac make up a divide-by-13 counter that generates the character scan address for the character generator and scans 0-12 on lines R0-R3. Adding U34a, U38, U37b, U39a makes the overall counter divide by 262. This part of the counter generates the line number for the memory on lines A5-A8. The counter counts line numbers 0-19. Lines 0-15 are displayed and 16-19 are blanked. This makes the vertical frequency 60.11 Hz (15750 ÷ 262).

R0 is a low for 63.5 μs and high for 63.5 μs, except that every sixth low will be 126 μs. R1 is low for 127 μs and then high for 127 μs, except that every third low will be 190.5 μs. R2 is low for 254 μs, high for 254 μs, low for 254 μs, and then high for 63.5 μs. R3 is low for 508 μs and then high for 317.5 μs. A5 is low for 825 μs and high for 825 μs. A6 is low for 1651 μs and high for 1651 μs. A7 is low for 3302 μs and high for 3302 μs, except that every third high is 825 μs. A8 is low for 9905 μs and high for 6603 μs. QD, the next binary digit, is low for 12.4 ms (15 lines) and high for 4.1 ms (5 lines). This becomes the vertical blanking signal. These signals repeat regularly for 260 rows, at which point two extra rows are inserted to make the total 262. U40 and U39bef pro-

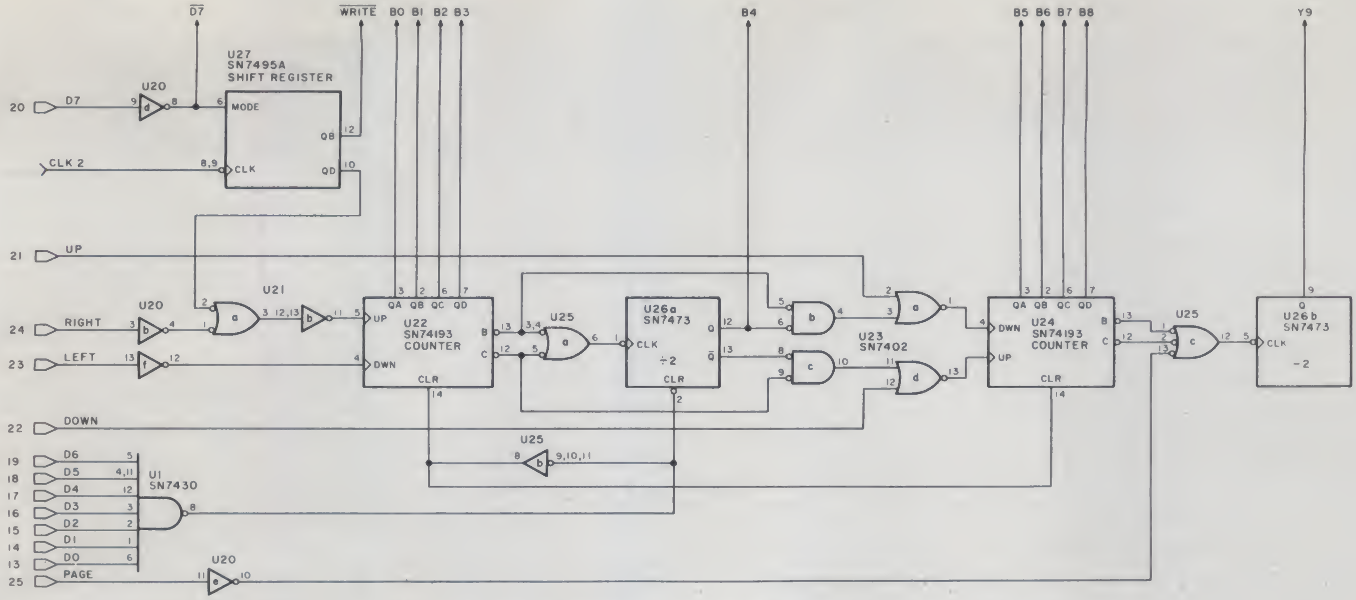


Fig. 6. Write circuits logic diagram.

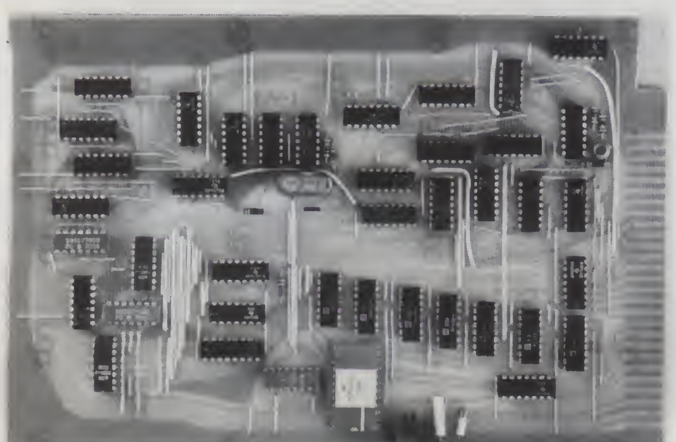
duce a low output during count 17 for 825 μs, the vertical sync pulse.

The write clock, shown in Fig. 6, acts as a register to store the current write address, the memory location into which the next character will be written, and generates the write signal that loads the character into that memory location. The current write address appears on lines B0 through B8 and Y9. There are seven ways in which this address may be changed.

1. If all seven input lines, D0-D6, are set high, U1 produces a low output that resets the counters to zero. This is the address of the upper left corner of the display.
2. If the D7 line is set high, U27, the shift register, will generate two pulses. The first is the write pulse, which is 661 ns long. This is followed, after a 661 ns wait, by a 661 ns count pulse that appears at pin 10 of U27. The count pulse advances the write clock by one character position.
3. If the "right" input is momentarily set high, the write clock is advanced by one character position.
4. If the "left" input is momentarily set high, the write clock is backed up by one character position.
5. If the "down" input is momentarily set high, the write clock is advanced by one line.
6. If the "up" input is momentarily set high, the write clock is backed up by one line.
7. If the "page" input is momentarily set high, the write clock is advanced by one page. If the display is on page two, it will advance to page one.

The data selector circuits are shown in Fig. 7. There are 19 input lines and nine output lines. Lines A0-A8 are the inputs for the read address from the horizontal and vertical sweep circuits. Lines B0-B8 are the inputs for the write address from the write clock. Lines Y0-Y8 are the outputs that deliver the selected address to the memory. The last input is the select line, the D7 line. When D7 is set high, indicating that a character is about to be written, the write address is selected and sent to the memory. When D7 is low the read address is selected.

The cursor generator, shown in Fig. 8, is used to generate a video signal that appears on the CRT as a line under the character position into which the next character entered will be written. U15, U18 and U17 make up a 9-bit comparator. The output lines will all be high only when the read address is



Completely assembled board.

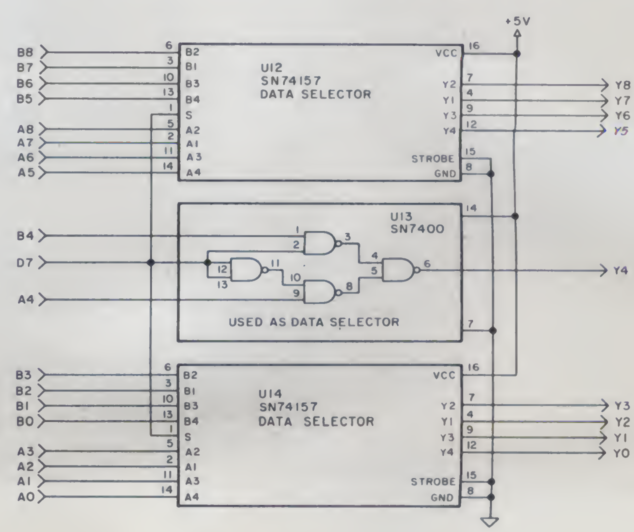


Fig. 7. Selector circuits.

equal to the write address. U16 combines the four lines from the comparators with the R3 and R2 lines to produce a low output only when the read address equals the write address and the CRT is scanning the bottom row of the character. The cursor signal out of U16 is then delayed by one character position by shift register U19 to compensate for the fact that the character being displayed is the character from the memory location immediately preceding the current read address. This is the result of the delay of about 500 ns between the time the read address is changed and the output of the memory changes.

The memory, shown in Fig. 9, is used to store the 1024 characters that make up the two pages. The 7-bit ASCII character codes on lines D0-D6 are continually presented to the memory input lines. However, the character is not written into the memory until the write line is set low. When this occurs, the character is written into the memory at the address present on lines Y0-Y9. The memory output is continually presented to the character generator, U10, and the output buffer, U3.

The character code present on these output lines is the character stored at the location defined by the address on lines Y0-Y9. The MCM6571A character generator, U10, accepts the character code on lines M0-M6 and the character scan row number on lines R0-R3 and sends the ap-

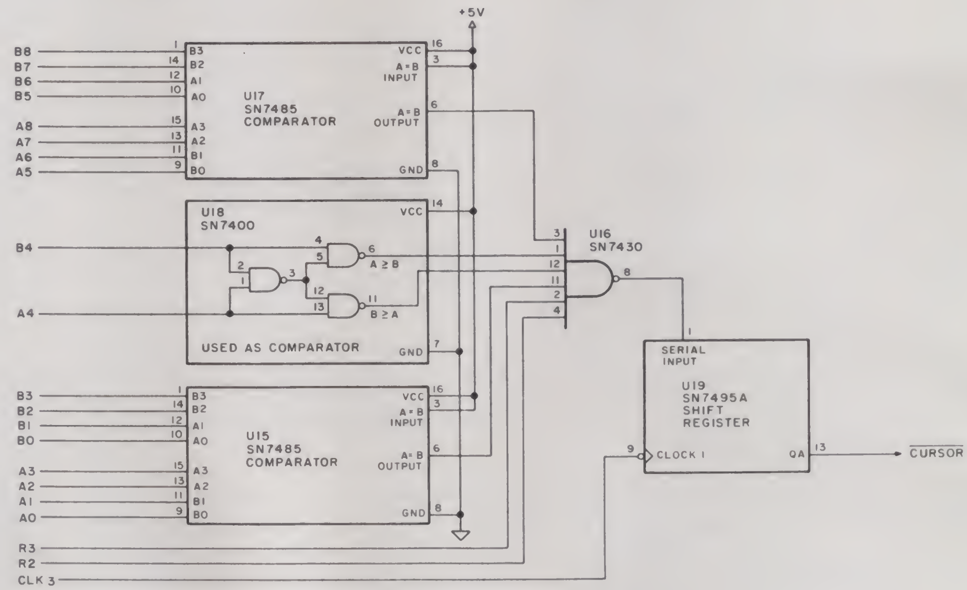


Fig. 8. Cursor control circuits.

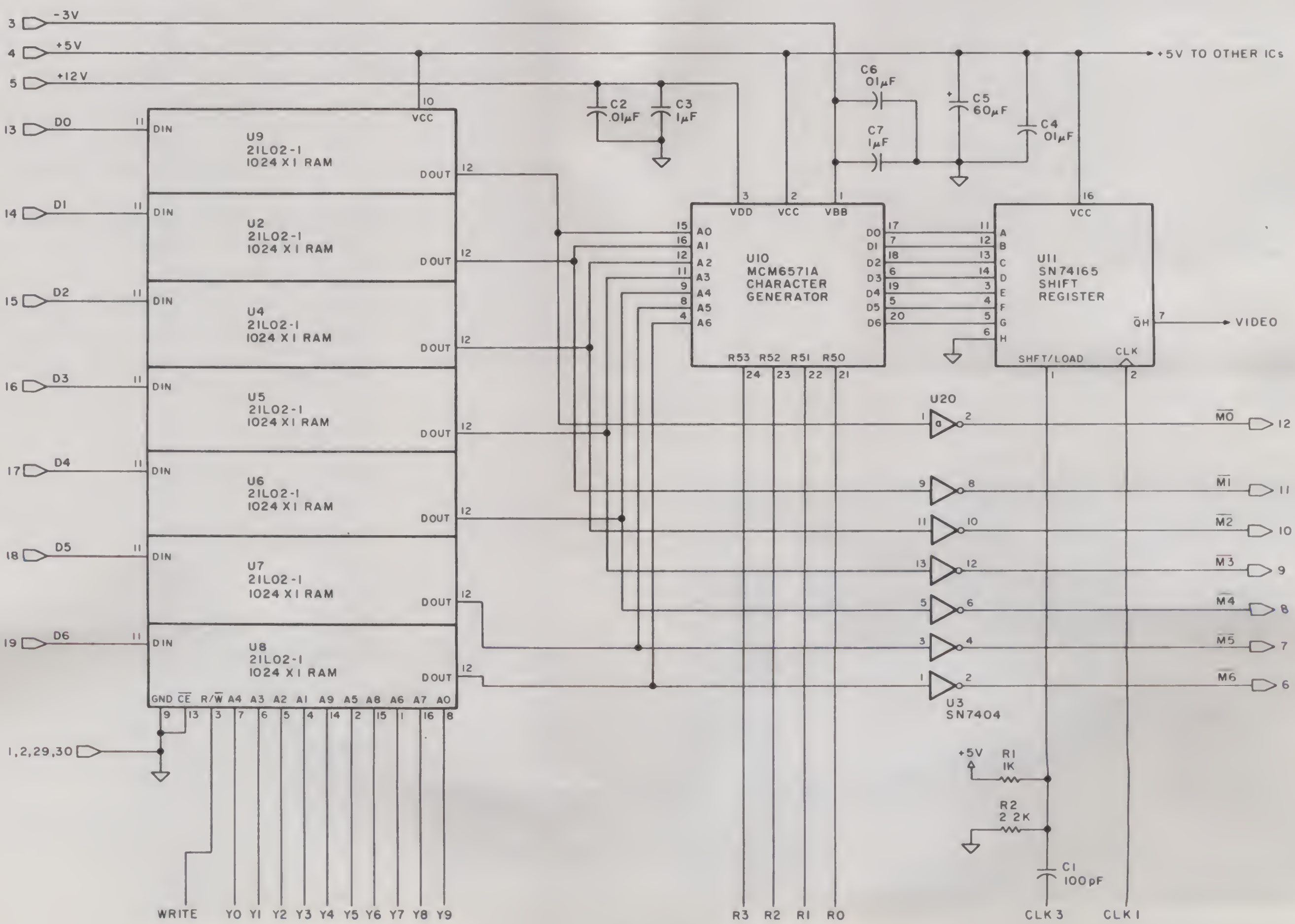


Fig. 9. Memory and character generator circuits.

ARE YOU TAKING FULL ADVANTAGE OF YOUR COMPUTER?

☐ **NO** Then we want you. Assembled at the Personal Computing Festival of the 1979 National Computer Conference will be scores of experts with diverse backgrounds. Your imagination cannot help but be triggered with new uses and applications for your own computer. Over 25 technical sessions will cover such topics as: languages, education, robotics, small business applications, speech synthesis and recognition, and investment analysis. Live demonstrations of applications by individual users will enable you to see the latest personal computing realities in action. Commercial exhibits will bring you up-to-date on industry offerings in microcomputer products, systems, and services. Take full advantage of your computer by attending the NCC Personal Computing Festival at the Americana Hotel in New York City, June 4-7. For continuing information about the complete computing experience at NCC, simply fill out and return the coupon below.

☐ **YES** Then we need you. You have developed an application to a point where your results are up to your own high personal standards. Speak to us, and the world, at the NCC '79 Personal Computing Festival. You can help your friends and colleagues take full advantage of their own computers by presenting a paper, chairing a session, or demonstrating your application. Valuable prizes will be awarded for outstanding applications demonstrations and for the best papers published in the *NCC '79 Personal Computing Proceedings*. Fill out the coupon below, check the box on conference participation, and we'll send you complete information. The deadline for submitting ideas and proposals is February 1, 1979. But remember...whether or not you're presently taking full advantage of your computer, there's a world of information on the very latest in personal computing awaiting you at NCC '79 in New York.

NCC '79 KB

PERSONAL COMPUTING FESTIVAL
c/o AFIPS, 210 Summit Avenue, Montvale, N.J. 07645

☐ Please keep me up-to-date on NCC '79 and its Personal Computing Festival.
☐ Please rush me information on participating in the Personal Computing Festival program sessions.
☐ I'm interested in demonstrating my own personal computing application; please send me details.
☐ My company is interested in exhibiting at the Personal Computing Festival.
☐ Please send information on the NCC '79 Travel Service.

Name ______________________ Title ______________

Company ______________________________

Street ______________________________

City ______________ State ______________ Zip __________

Reader Service—see page 179

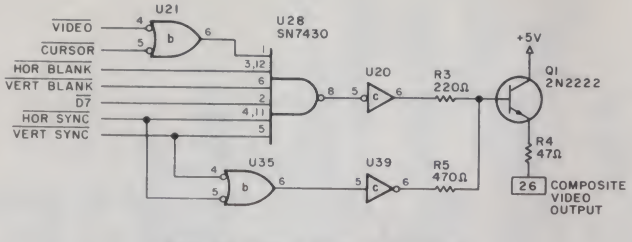


Fig. 10. Video output circuits.

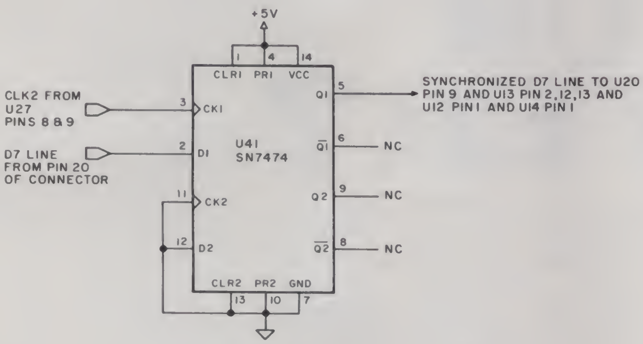


Fig. 11. Strobe synchronizer.

propriate seven bits to the shift register, U11. The 756 kHz shift register load pulses are a differentiated square wave riding on a dc bias of 3.4 V. The RC time constant is about 0.07 μs. When the shift register load input is momentarily set low, it loads these seven bits, plus an eighth zero bit (for character spacing), in parallel and begins sending them out on the video line—one bit for each low-to-high transition of the shift-register clock input.

The video section is shown in Fig. 10. U21b combines the video and cursor signals so that a low from either one produces a high output, which represents a white level on the CRT. U28 and U20c combine the video signal from U21c, the horizontal sync, the vertical sync, the horizontal blanking, the vertical blanking and write blanking (D7). A high out of U20c represents a white level on the CRT. This results only when all inputs to U28 are high.

The presence of a sync or blanking pulse (low state) produces a high out of U28, which represents a black level on the CRT. U35b and U39c combine the sync lines. The presence of either a horizontal or a vertical sync produces a low out of U39c. This can occur only during a blanking pulse. Q1 and its associated resistors combine the video and the blanking signals with the sync signal to give three-level composite video output signal.

During the sync pulses, both U20c and U39c will be low, resulting in zero output voltage. If a sync pulse is not present, U39c will be high. Then U20c can produce a black or white level as required by the video and blanking signals. A white level is about 3.5 V out (open circuit), and a black level is about 1.1 V out.

A Correction to the TV Readout Circuit.

It was intended that the shift register U27 would generate the write and count pulses at a rate determined by CLK2 whenever the D7 line was set high. However, if the transition of D7 from low to high occurs at exactly the same time as the transition of CLK2, the shift register U27 will not generate the write and count pulses. When this happens, that character will not be written into the memory. This occurred about three or four times out of 100 characters.

This loss of characters was completely eliminated by the addition of the strobe synchronizer, shown in Fig. 11. It consists of a single IC, U41, a D-type flip-flop. This serves to synchronize the D7 transitions with the CLK2 signal so that the signal sent to shift register U27 always changes state just after a transition of the CLK2 signal, thus preventing the loss of characters.

U41 can be added to the PC board to the left of U28 and below U19 (see Fig. 12). Its installation requires cutting only the line coming from pin 20 of the edge connector. U41 is inserted in this line so that the D7 signal must pass through it before going into the rest of the circuit. The jumper wires required to connect U41 to the circuit can be easily located by comparing Fig. 1 with Figs. 11 and 12. The photos are also helpful.■

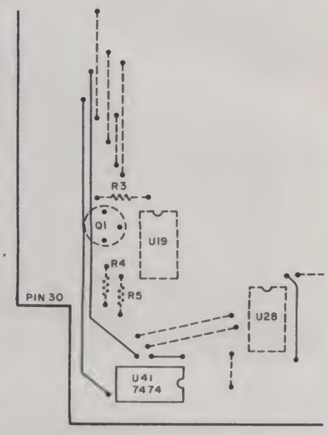


Fig. 12. Mods for installing strobe synchronizer.

C1	100pf 10V	
C2, C4, C6	.01uf 20V	
C3, C7	1.0uf 20V	
C5	120uf 10V	
Q1	2N2222A	
R1	1K ¼W	
R2	2.2K ¼W	
R3	220 ¼W	
R4	47 ¼W	
R5, R6, R7	470 ¼W	
U1, U16, U28, U40	SN7430	8-input NAND
U2, U4, U5, U6, U7, U8, U9	MCM21L02-1	1024-bit static memory
U3, U20, U31, U39	SN7404	hex inverter
U10	MCM6571A	character generator (16K ROM)
U11	SN74165	8-bit shift register
U12, U14	SN74157	quad 2-line-to-1-line selector
U13, U18, U21, U29, U35	SN7400	quad 2-input NAND
U15, U17	SN7485	4-bit magnitude comparator
U19, U27	SN7495A	4-bit shift register
U22, U24	SN74193	4-bit up/down counter
U23	SN7402	quad 2-input NOR
U25, U37	SN7410	triple 3-input NAND
U26, U34	SN7473	dual J-K flip-flop
U30, U32, U36, U38	SN7493	4-bit counter
U33	SN7420	dual 4-input NAND
X1		24-pin socket for U10 (if desired)
XTAL		6048 kHz crystal
PCB		printed circuit board DVA-1

Table 1. Parts list.

TRS-80 software

Orcomp ✓011

ORCOMP DATA SYSTEMS
P. O. Box l357
Beaverton, Oregon 97005

Write us today for our complete brochure of ready to use TRS-80 programs.
Or, just circle our bingo number.

Percom's LFD-400 Floppy Disk System

If your microcomputer is not based on the S-100 bus, but, rather is based on the SS-50, don't feel half-human. This system gives maximum utility—and not at a giant-sized price.

Robert R. Wier
1208 Mistletoe Dr.
Fort Worth TX 76110

For those hobbyists who own a microcomputer in which the bus structure is not S-100, the announcement of a new product for their bus is a particularly noteworthy event. So, when the Percom Data Company announced a floppy-disk system based on the SS-50 bus, which is characterized by the SWTP M6800 computer system, considerable interest was shown by a number of Southwest owners at the local computer club. Harold Mauch of Percom was invited to speak and he gave an interesting talk on his design philosophy concerning the new disk system. I was impressed with the features of the unit he described and acquired one in short order.

The LFD-400 minimum configuration, which I purchased, consists of a Shugart SA-400 minidisk drive, a controller board that plugs into one of the "large" slots on the SS-50 bus and associated power supply, cables and cabinet. On the controller board is a preprogrammed 2708 EPROM that contains MINI-DOS[1], Percom's minimal disk-operating system. The controller board also contains space for another 2K of EPROM with a view toward a more comprehensive DOS (disk-operating system) for future expansion.

Of course, you could store any program you wanted to retain permanently in the 2708s. Special voltage regulators are included for the +12 V line, which is somewhat marginal in the original M6800 system. An inactivity time-out is included that shuts off the drive motor if the disk is not accessed for a period of time in order to preserve motor life in the drive. A proprietary circuit is included to counteract interior track "bit shifting" effects and increase reliability. The price class is $600 assembled and tested.

The LFD-400 system consisting of controller board, drive, power supply, cables and cabinet.

My decision in purchasing this particular floppy-disk system was rooted in two considerations:

1. My system at the time only contained 4K of RAM.
2. Low cost was essential.

You may, perhaps, wonder why I purchased a floppy disk before I added more memory. Well, on my system the programming consists largely of real-time applications (such as ham radioteletype). This means that languages such as BASIC are not generally suitable since software timing loops are employed and the programs must execute *fast*; this may be difficult with an interpreter such as BASIC. Also, the chief (and only) programmer (me) *likes* to program in assembly language and has not found 4K to be very limiting.

Directory or Do-It-Yourself?

One of the problems with many disk systems, though, is that the DOS requires a certain amount of RAM in which to function. This will typically run from 8 to 16K, which obviously rules out their use with a 4K system such as mine. However, the minimal Percom system requires only 32 *bytes*, plus whatever amount of space your program occupies. This is accomplished by having the DOS in the 2708 EPROM on the controller board, while the first 32 bytes of memory are used for calculated quantities such as checksums and so forth.

Of course, there is a trade-off involved with this small appetite for memory. The disk-operating system is quite restricted in comparison to some of the

full DOS's appearing in other floppy systems. But there are several reasons (besides low-memory requirements and small price) that a home user can live with the functions provided by MINI-DOS.

At the most basic level the LFD-400 is used as essentially a cassette replacement. MINI-DOS allows data to be easily saved and retrieved from the disk as specified by the user. The organization is such that the disk appears to be 350 contiguous sectors, and there is no need to worry about specific track numbers. There is no directory on the disk itself, since the user *writes down* on paper the addresses for the saved data on the disk (indeed, as he probably does for cassette). I've found the *K&E Field Book*, which is about 5 by 7 inches with ruled, columnar paper, to be ideal.

While at first the use of the "manual" method may seem like heresy to those connected with computing in a professional capacity, there exist some valid reasons for using it.

1. A directory takes up space on the disk that could be used for data. Since the object is to provide a disk system of maximum utility at minimum cost, why not have the user take over this responsibility (i.e., trade user time for hardware/software cost)?

2. A directory occupies a critical position of importance in the disk system. It will be accessed frequently, since every time a file is to be found, the directory must be read. Note also that since the directory is heavily used, it is also the most prone to damage. Should the directory be unusable through damage or misuse, the programmer cannot easily find out which files are present and where they are stored. Recovery from this situation could be most frustrating. In contrast, a manual directory is probably less subject to catastrophic destruction, although the user may be tardy in updating entries (although this only happened to me once—I learn fast!).

3. A directory must be read into memory in order to search it for file names and locations. The longer the directory is, the more RAM is required to hold it during a search.

On the other hand, if you do want more sophistication, Percom offers MINI-DOS +[1]. MINI-DOS + is an extension of MINI-DOS that does keep a disk directory and allows up to 31 named files per disk. The names may be such that they indicate a write-protected file so that you cannot easily destroy a file by accident.

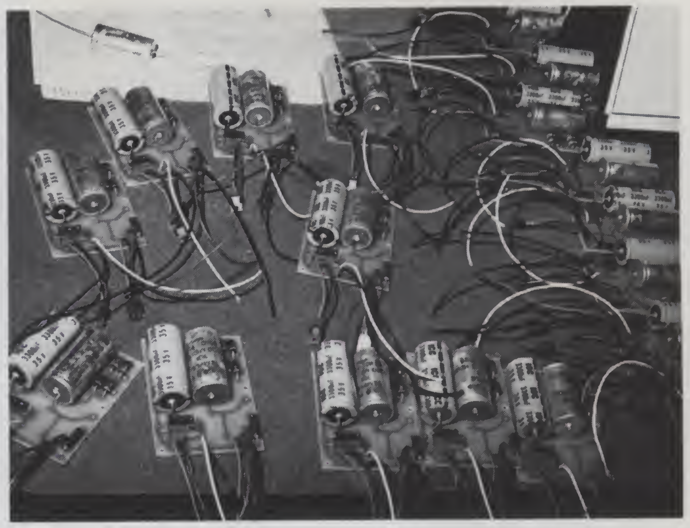
A number of convenient commands are available for listing the directory and running a program by a command such as RUN BASIC. This causes the program to be loaded and a jump executed to the first instruction in one operation. Like MINI-DOS, MINI-DOS + is contained in EPROM, so that the additional RAM required is minimized.

If future expansion were planned, a minimal system such as the LFD-400 could easily be integrated into the overall configuration. If full-sized 8 inch disk drives were acquired, with their improved access speeds and data capacity (and, of course, greatly increased cost), the LFD-400 could contain the DOS for the large drives.

Operation

After I acquired an LFD-400, I took it home—then in the country—and plugged it into my SWTP system. Living 100 miles from *anywhere* necessitated a high-reliability factor, since running down to the nearest computer store involved a four-hour drive. That was the reason (along with low initial cost) for selecting the M6800 in the first place. The Shugart SA-40 drive is widely used proven reliable.

When power was first applied to the LFD-400, lights lit and the drive responded when the RESET button on the computer was pressed. Entry into MINI-DOS is accomplished through the Z command if you have SWTBUG[2] or a jump instruction otherwise (your SS-50 system should have some type of MIKBUG[3] equivalent



No, not Kingdom of the Spiders. *Power supplies assembled and ready for testing.*

(SWTBUG, SMARTBUG[4], etc.) since MINI-DOS uses some of the MIKBUG I/O routines).

With MINI-DOS, to load data from the disk into memory, simply specify which sector to start with and whether you want the data loaded back into the same position in RAM that it was saved from. Optionally, you may specify to load into a different location. MINI-DOS will then load data starting with the sector specified and continue loading until an end-of-file marker is sensed in the data on the disk.

Saving data is almost as simple. You specify the memory starting and ending addresses and the sector on the disk to start with. MINI-DOS will then tell you how many sectors were used, and you enter the ending sector number into your "manual" directory.

Since Percom supplies patches to use with some of the better known M6800 programs, I decided to try to load one as a test. I received an error message from MINI-DOS saying there was no disk in the drive. I pulled the disk out, put it back in and received the same error again. Unless I've been breathing soldering iron fumes too long, there *is* a disk in the drive. Well, back to the instruction book.

The write-protect slot in the disk should be *toward* the LED. Success! Nothing like putting the disk in upside down to confuse the issue (and the drive). Fortunately, the Shugart drive was forgiving of my ignorance, and no data was lost. The disks used are the easily available ten-sector, hard-sectored "one-sided" mini-diskettes. Each disk will carry about 90K bytes and costs about $4.

Improvements

As I became familiar with the operation of the LFD-400, some previous problems with I/O essentially disappeared. In my own personal computing system, I've noticed that with the addition of certain pieces of hardware, the utility of the system is greatly improved. In fact, the improvements in the ease of use and in the ability to successfully tackle large software projects have been so dramatic that they might be called "breakthrough" points.

The first of these occurred with the transition from the use of paper-tape I/O to the use of magnetic-tape cassette. This improvement could be characterized as an improvement in the physical aspect of program storage, what with the elimination of the tiresome handling of paper tape, which is fragile, oily and prone to read errors.

Another dramatic improvement happened with the addition of the LFD-400. In this case the dramatic improvement occured in the temporal sense, with I/O times being cut to a small fraction of those existing

Harold Mauch of Percom at the keyboard. A three-drive version of the LFD-400 is underneath the M6800 computer to the right of the ADM 3A.

with cassette tape. This is the reason that a floppy disk appears much closer to an extension of your system's RAM than to your serial storage capacity. Some of these improvements are nonobvious.

For example, the first time any assembly-language program is run it will most likely go off into never-never land and destroy itself, thereby erasing the evidence of what caused the malfunction in the first place. The only way to debug something like this is to load the program in again and set breakpoints at (you hope) logical places in the execution sequence so you can ascertain which part of the program is suicidal. If you pursue this method with a long program on paper tape or even cassette, you spend more time loading the program than debugging it.

Not so with the floppy disk. If the program bombs itself, you simply load it back into memory in a few seconds and set the breakpoint earlier. Computing is less complicated and more rewarding when you use the floppy disk. Another nice thing is that when patches are put into a program, each successive version of the patched program may be easily saved so that if one patch creates more problems than it solves (a not infrequent occurrence), you simply load the program saved one patch previously and try something different.

Certain classes of programs require the use of a floppy or some similar media for practical use. Any program that requires a previously built data base benefits greatly from the increased access speed of a floppy disk. Things such as text editors and mailing-label programs may use tape, but they are so much easier to work with if the data is readily accessible.

Also, some game programs benefit from ready storage. For example, you might wish to save a Star Trek game to resume playing later. Or, in many artificial intelligence programs, it is necessary to save the program's "knowledge" or it will be necessary to teach it all over again the next time it is loaded.

MINI-DOS is supplied with a source (assembly) listing so that if you are inclined to write your own disk routines, you may know where to access the save and load portions of MINI-DOS in EPROM and thereby significantly decrease the I/O memory requirement and complexity of your programs. The LFD-400 is simple enough for the novice to use (indeed, probably more so than some of the more advanced DOS's), but the hardware is sufficiently "visible" to the expert so that he may easily go and program device control.

Harold Mauch calls this "feeling close to the bytes." There is a certain feeling of satisfaction in being able to make the disk arm move back and forth successfully if you've never done it before.

An active user's group is forming, and Percom is sending out a newsletter containing contributions concerning various topics on the LFD-400. For those who wish a configuration beyond MINI-DOS+, more advanced DOS's are being developed. The controller board can support up to three drives. Patches are available for BASIC to allow chaining of files from one disk to another, thereby allowing quite large programs to be executed through overlaying.

Anyone using the SS-50 bus would do well to consider the LFD-400. It provides maximum utility at a minimum price. If you are interested in obtaining a floppy-disk system that doesn't cost an arm and a leg, the Percom LFD-400 clearly merits serious consideration. ■

---

1. MINI-DOS and MINI-DOS+ are trademarks of Percom Data Company.
2. SWTBUG is a trademark of SWTP.
3. MIKBUG is a trademark of Motorola, Inc.
4. SMARTBUG is a trademark of Smoke Signal Broadcasting.

Reader Service—see page 179

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

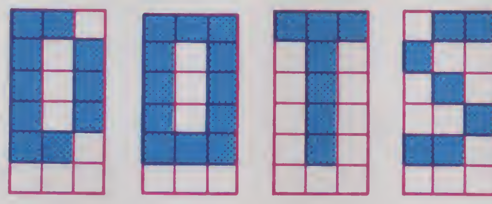
FIRST CLASS PERMIT NO. 17 PETERBOROUGH NH 03458

POSTAGE WILL BE PAID BY ADDRESSEE

kilobaud
MICROCOMPUTING

Peterborough NH 03458

Tom Pittman
PO Box 23189
San Jose CA 95153



Anyone interested in software character generation should read "Dots." Part 1 deals with the program logic and philosophy of operation—and gives you a chance to make $50.

Many of the early graphics-display systems for microcomputers (the Cromemco Dazzler comes immediately to mind) had fairly coarse resolution and did not come with built-in character generators. Software-generated text characters come out relatively large, and not many will fit on a screen at once. Rumors of new video-graphics systems with much higher resolution (using, of course, much more memory) abound, but these may still need software character generators.

My recent experience with an extreme case of low resolution with software character generation may be of some help to those with similar problems. And I think that the techniques explained in this article will also help those who are trying to get more text on a screen with otherwise reasonable resolution.

The article is divided into two parts. Part 1 deals with the program logic and the philosophy of operation and will be valuable to anyone who is interested in the concept, whatever microprocessor he uses. Part 2 will deal with the specific routines to implement a software character generator in the RCA 1802, where the need for this algorithm is most acute.

Determining Character Size

When Netronics Research in Connecticut asked me to prepare a version of Tiny BASIC to run on the Elf II, an inexpensive 1802-based computer, the requirement also included putting the ASCII output on the video display, which is an integral part of the computer. Unfortunately, the video interface is low resolution, with only 64 pixels (picture elements) in the horizontal direction. The display circuit is capable of 128-pixel resolution in the vertical direction, but RCA (who manufactures the video display chip) generally recommends a software fix-up that results in approximately square pixels in a 64 × 32 grid. This is what I had to put text on.

One possibility is to force everything into three pixels wide. This runs into a conflict in the representation of "M," "W," "H" and "N." With two full-length vertical bars on the outside, there are only five positions for a single dot in the middle, which may each be assigned to one letter, as shown in Fig. 1a. This gives a recognizable, albeit odd, representation for all letters. Other possibilities for "M" and "W" in a 3 × 5 cell are shown in Fig. 1b.

A uniform cell width of four pixels is not even to be considered, since there are too many characters that are symmetrical about a single vertical element.

I chose a different approach. I observed that many characters can be quite satisfactorily represented in a width smaller than three pixels. Most notable are the punctuation symbols ! . , ( ) ' : and so on. But also some letters are tolerable in narrower widths. More significantly, spaces need not be a whole character wide if we allow variable spacing.

Once I opened this can of worms, most of the size problems went (wriggled?) away. I would represent each character in the minimum width cell to make it recognizable. "M" and "W" would take five pixels (cell width 6); most letters and all digits would be three pixels wide (cell width 4); spaces, periods and colons would be one pixel wide (cell width 2); and some of the other characters might be two or four pixels wide. While I could have fit the numeral "1" into one or two pixels, I chose not to, so that it would be easier to get even columns of numbers.

Since this was to be done in software, I decided there was no reason to restrict myself to a height of five pixels. Aesthetics required that all characters occupy a uniform-height cell (six pixels including line spacing), but if I did not go to excess, I could allow descenders on some of the characters. This helped make the "Q" more readable and avoided the disgusting appearance of a comma resting halfway up the line.

A 5 × 7 dot matrix was out of the question. It would require a minimum character cell of six pixels wide by eight high, of which I could fit four rows of ten characters on a line. With all the gripes I hear about 16 × 32, 4 × 10 would be unacceptable. Besides, a typical Tiny BASIC program line averages 17 characters. I decided to analyze what actually constitutes a minimum character cell.

The characters, "E," "B," "3," "8," "S," "5" and so on, with three distinct horizontal members, require a minimum cell height of five pixels: three for the horizontal members and two blank pixels to separate them. This is clearly a minimum for these characters, and there are enough of them that it is not reasonable to consider them to be exceptions. There are a few special characters such as "$," "%" and "@" that could be considered to have more than three distinct horizontal members, but I chose to compromise on the grounds of their infrequent use.

There is a great number of characters with two distinct vertical members, but only two ("M" and "W") have three. The existence of these two has historically forced the industry to consider five pixels the minimum width. But since five pixels was already demonstrably unacceptable, I sought another solution.

Having elected to support descenders, I proceeded to lowercase, which was an easy next step. Even as I write this article I am struggling with a (older) text-processing system that displays only uppercase on the screen (though lowercase is stored in the buffer). As more and more computers are used in text-processing applications, more and more we will see the need for full uppercase and lowercase text.

Alas! This required a few more compromises, though fortunately not like those of terminal manufacturers who display lowercase without descenders. In my opinion, the use of a reduced-size capital is an inexcusable substitute for a lowercase "p," and I did not even consider lowercase an option worth persuing until I had already allowed descenders.

The problem had to do with the number of distinct horizontal elements in certain of the

lowercase letters. The worst problems were "e," "a" and "z." By setting the normal (capitals) cell height to six, these lowercase letters were obliged to fit into three vertical pixels.

Our Roman alphabet normally uses a three-horizontal-element representation for the letter "a" (as you see in this magazine), but certain Gothic and italic fonts have a two-element representation, so I used that instead. Fig. 2a illustrates the font choice and the representation chosen.

The letter "z" is infrequently used, so I compromised on its shape: full-width top and bottom horizontal elements define the letter outlines, and a single pixel connecting them in the center represents (but does not look much like) the diagonal bar. This is illustrated in Fig. 2b.

The hardest problem was to represent the most common letter, "e." I fudged. It does not look a lot like an "e," but once you get used to it, it is not too bad. In this particular representation its outline approximates the outline of a lowercase "e." This is illustrated in Fig. 2c.

The whole set of 94 ASCII graphics, plus space and DEL, is shown in Fig. 4. You will notice a few oddities, for which I offer no excuse other than the constraints of the project.

Let's face it: How can you make a recognizable ampersand ("&") only five pixels high? And there was no way I could make an asterisk ("*") come out anything but a blob if I put more than four arms on it. So it looks like a raised "x." The dollar sign ("$") looks better from a distance.

The Back Space

The next problem to face is the back space. If each character is a different width, how do you know how far to back up? . . . or how much to erase from the display buffer? Rather than trying to define a special command to erase the character pointed to, I elected to let the back-space character also erase the character backed up over. Separate cursor controls can give a nondestructive positioning capability.

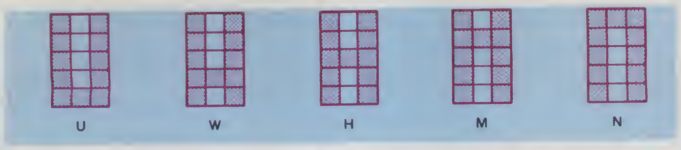


Fig. 1a. Five letters with two vertical bars connected by a single dot.

Fig. 1b. Alternative representations for M and W in a 3 × 5 matrix.

Besides that, either the generator program must clear out the space it is going to put the next character in or it must assume it is already clear. The latter case is easier to implement because scrolling will generally clear the new line.

I admit that ease of implementation is a crummy reason for making a program design choice, but in this case the ability to overstrike seemed to be a good option to leave open (for possible APL implementations).

But the question remained: how far to back up and erase? I settled on separately stacked pointers to the last character.

The program maintains a little stack of numbers encoding the low three bits of the horizontal position (pixel number) of the left edge of each character. There is no logical reason why I could not have saved the whole horizontal position. It simply turned out that the low three bits were separately controlled, and three bits were enough. No character is more than six pixels wide, so if the saved pointer is greater than the current position, subtract another 8.

The reason the low three bits are separated is that in this hardware each eight pixels on one line are packed into a single byte. Thus the word pointer was in an address field (two bytes), and the bit position within the selected byte was in a separate byte. It is strictly a matter of convenience, but I think it may have improved performance by minimizing the packing and unpacking of the pointer.

It behooves a stack not to grow too large. Therefore, I (rather arbitrarily) dump the stack whenever a cursor-changing control character (other than back space, of course) is processed. This also saves me the bother of stacking a full cursor position. Carriage returns, line feeds and form feeds all reset the stack to Empty.

Then, just in case the user is dumb enough to type excessively long lines with no carriage returns, I arranged the stack to overflow into the display buffer. Gotcha! If the stack overflows, the user starts to see little pieces of trash at the top of the display. If the abuse continues, the stack is forced to a reset when it reaches a page boundary.

Ideally, I suppose, a circular buffer could have been used. Then the last *n* characters could be backed up, regardless of how long ago the last carriage return occurred. The problem is to pick a reasonable *n*.

Storing The Characters

Another problem to be resolved was the representation of the dot matrix information in the program memory. That the buffer memory was organized as eight horizontal pixels to a byte dictated that each character would be stored as six rows of dots, one row to a byte. Six bytes times 96 characters is 576, which seems like a lot of memory. Many of the characters have common features, and with a few hours of juggling and packing I was able to fit all 96 characters into 143 bytes.

This requires a second table in ASCII sequence to point into this table and to identify which part of the byte the character is sitting in and how big it is. This can be accomplished with two bytes per entry, for a total of 192 + 143 (335) bytes to store all the characters.

Now that I think about it, I am not so sure that the extra complication in the program did not use up most of the 241 bytes I saved in the tables. Fig. 3 illustrates a *small* part of the character table.

Each entry of the pointer table is, as I said, two bytes. There is no particular reason for ordering them one way or the other, but for convenience I will refer to the ordering of my program. The first byte contains two fields. The leftmost five bits are a mask to serve the double purpose of signifying the size of the character and to be used to isolate it from the neighboring bits in the dot table.

I could have used a three-bit numerical value here, but then the program would have had to generate the mask. This saves a step. The low three bits indicate the position of the dots for this character within their respective bytes. It is a count of how much to the right the character is shifted from a left-justified position.

Fig. 2a. Roman and italic minuscule "A," with dots representation in a 3 × 3 matrix.

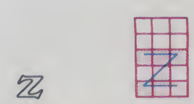


Fig. 2b. Dots representation of minuscule "Z" in a 3 × 3 matrix.

Fig. 2c. Approximation to minuscule "E" by dots in a 3 × 3 matrix.

DSI COMMUNICATIONS SERIES

1.3GHz — 1GHz — 700MHz

MODEL C1000 10Hz to 1GHz

$499.95

- INCLUDES BATTERY PACK
- AUTO ZERO BLANKING
- AUTO DECIMAL POINT
- 10MHz TIME BASE

Accuracy . . . that's the operational key to this rugged advanced design Model C1000 1GHz frequency counter . . . a significant achievement from DSI. That's because you get . . . **.1 PPM** 0° to 40°C proportional oven time base . . . Built in 25DB preamplifier with a 60DB adjustable attenuator . . . x10 & x100 audio scaler which yields .01 Hz resolution from 10Hz to 10KHz equivalent to 10 sec. & 100 sec.Gate Time . . . Selectable .1 & 1 sec. time base and 50 ohms or 1 meg ohm input impedance . . . Built-in battery charging circuit with a Rapid or Trickle Charge Selector . . . Color keyed high quality push button operation . . . All combined in a rugged black anodized (.125" thick) aluminum cabinet. The model C-1000 reflects DSI's on going dedication to excellence in instrumentation for the professional service technician, engineer, or the communication industry.

MODEL C700 50Hz to 700MHz

$369.95

- INCLUDES BATTERY PACK
- AUTO ZERO BLANKING
- AUTO DECIMAL POINT
- 10MHz TIME BASE

ALL NEW! All UNPARALLELED DSI QUALITY! The model C 700 700 MHz frequency counter features . . . **.2 PPM** 0° to 40° C proportional oven time base . . . 25db preamplifier with a 60db adjustable attenuator. Built in battery charger with a rapid or trickle charge selector . . . Combined in a rugged (.125" thick) aluminum cabinet makes the C700 ideal for the communication industry and professional service technician.

3600A OWNERS: Up date your 3600A frequency counter to a C 700 includes, new back board, **.2PPM** proportional oven, 25db preamplifier, rugged .125" thick aluminum cabinet, order 3600A-700. Unit must be returned to DSI factory for modification.

DSI — GUARANTEED SPECIFICATIONS — FACTORY ASSEMBLED — MADE IN USA

Model	Frequency Range	Proportional Oven Accuracy Over Temperature	50Hz To 75MHz	75MHz To 500MHz	500MHz To 1GHz	Number Of Digits	Size Of Digits	Power Requirements	Size
C700	50Hz to 700MHz	.2PPM 0° to 40°C	50MV	10MV	NA	8	.5 Inch	115 VAC-BATT 8 to 15VDC	3"H x 8"W x 6"D
C1000	10Hz to 1GHz	.1PPM 0° to 40°C	20MV	1MV	>50MV	9	.5 Inch	115VAC-BATT 8 to 15VDC	4"H x 10"W x 7½"D

— All Units Are Factory Assembled, Tested And Carry A Full 5 Year Limited Warranty —

FREE

Strongest warranty in the counter field.
Satisfaction Guaranteed.

FOR MORE INFORMATION

Call Toll Free: (800) 854-2049
DSI INSTRUMENTS, INC.
California Residents, Call Collect: **(714) 565-8402**
VISA • MC • AMERICAN EXPRESS • CHECK • MONEY ORDER • COD
7914 RONSON ROAD, #G, SAN DIEGO, CA 92111

Model C 700	**$369.95**
3600A-700 Factory Update (3600A only) Includes Labor & Re-Calibration	**$199.95**
Model C 1000	**$499.95**
Opt. 01 1.3 GHz (C1000 only)	**$ 99.95**
Opt. 02 .05 PPM 10MHz Double Oven 0° to 50° C Time Base (C1000 only)	**$129.95**
Ant. 210 Telescopic Ant./BNC Adapter	**$11.95**

Referring to Fig. 3, you notice that the "3" is three pixels wide and that it is shifted one position to the right of the leftmost bit in the bytes; its mask byte is therefore "11100,001," or E1 hex. The "8" in the same bytes begins three bits to the right of leftmost, so its mask byte is "11100,011," or E3 hex.

Apostrophe (') uses two of the lower dots in the "3" (one of which is also common to "8") and is only two pixels wide, starting two over from the left edge; its hex code is C2 for a binary pattern "11000,010."

The second byte in the pointer table entry is a relative offset from the beginning of the dot table. It points to the bottom byte of the six bytes required for the character. The next five bytes are successively the rows above this bottom row.

The reason I chose to start at the bottom is that the cursor normally sits at the bottom of the character cell, so that is a natural place to begin storing bits in the buffer. But there is again no logical reason why it could not have gone the other way. Tables 1 and 2 list the contents of the pointer and dot tables, respectively.

Space-saving Tips

One oddity of this program relates to the particular implementation and is not an essential ingredient. Because there are so few pixels for characters, I allowed the program to cut a character and split it between lines. It looks a little strange, but it does squeeze another half character onto the line (on the average). I think if I

ASCII	HEX	TABLE HEX	MASK	POSN	POINTER	ASCII
	20	8608	10000	6	8	
!	21	820A	10000	2	10	!
"	22	E508	11100	5	8	"
#	23	F823	11111	0	35	#
$	24	E435	11100	4	53	$
%	25	E55A	11100	5	90	%
&	26	F423	11110	4	35	&
'	27	C200	11000	2	0	'
(	28	C111	11000	1	17	(
)	29	C211	11000	2	17	)
*	2A	E03C	11100	0	60	*
+	2B	E547	11100	5	71	+
,	2C	C307	11000	3	7	,
-	2D	C441	11000	4	65	-
.	2E	8407	10000	4	7	.
/	2F	E529	11100	5	41	/
0	30	E111	11100	1	17	0
1	31	E243	11100	2	67	1
2	32	E44F	11100	4	79	2
3	33	E103	11100	1	3	3
4	34	E01D	11100	0	29	4
5	35	E042	11100	0	66	5
6	36	E249	11100	2	73	6
7	37	E073	11100	0	115	7
8	38	E303	11100	3	3	8
9	39	E049	11100	0	73	9
:	3A	8304	10000	3	4	:
;	3B	C335	11000	3	53	;
<	3C	E541	11100	5	65	<
=	3D	E503	11100	5	3	=
>	3E	E517	11100	5	23	>
?	3F	E05A	11100	0	90	?
@	40	E079	11100	0	121	@
A	41	E12F	11100	1	47	A
B	42	E56D	11100	5	109	B
C	43	E417	11100	4	23	C
D	44	E217	11100	2	23	D
E	45	E560	11100	5	96	E
F	46	E534	11100	5	52	F
G	47	E24F	11100	2	79	G
H	48	E33B	11100	3	59	H
I	49	E117	11100	1	23	I
J	4A	E155	11100	1	85	J
K	4B	E060	11100	0	96	K
L	4C	E343	11100	3	67	L
M	4D	F96D	11111	1	109	M
N	4E	F41D	11110	4	29	N
O	4F	E017	11100	0	23	O
P	50	E573	11100	5	115	P
Q	51	E00B	11100	0	11	Q
R	52	E53B	11100	5	59	R
S	53	E029	11100	0	41	S
T	54	E066	11100	0	102	T
U	55	E21D	11100	2	29	U
V	56	E379	11100	3	121	V
W	57	FB2E	11111	3	46	W
X	58	E260	11100	2	96	X
Y	59	E273	11100	2	115	Y
Z	5A	E035	11100	0	53	Z
[	5B	C017	11000	0	23	[
\	5C	E329	11100	3	41	\
]	5D	C117	11000	1	23	]
^	5E	E200	11100	2	0	^
_	5F	E507	11100	5	7	_
`	60	C300	11000	3	0	`
a	61	E184	11100	1	132	a
b	62	E07F	11100	0	127	b
c	63	E27F	11100	2	127	c
d	64	E30C	11100	3	12	d
e	65	E512	11100	5	18	e
f	66	E566	11100	5	102	f
g	67	E050	11100	0	80	g
h	68	E356	11100	3	86	h
i	69	8107	10000	1	7	i
j	6A	C007	11000	0	7	j
k	6B	E50C	11100	5	12	k
l	6C	850C	10000	5	12	l
m	6D	FB84	11111	3	132	m
n	6E	E384	11100	3	132	n
o	6F	E57F	11100	5	127	o
p	70	E575	11100	5	117	p
q	71	E13D	11100	1	61	q
r	72	C689	11000	6	137	r
s	73	E589	11100	5	137	s
t	74	E268	11100	2	104	t
u	75	E489	11100	4	137	u
v	76	E289	11100	2	137	v
w	77	F889	11111	0	137	w
x	78	E03D	11100	0	61	x
y	79	E555	11100	5	85	y
z	7A	E47F	11100	4	127	z
{	7B	E011	11100	0	17	{
|	7C	8002	10000	0	2	|
}	7D	E211	11100	2	17	}
~	7E	E500	11100	5	0	~
	7F	F203	11110	2	3	

Table 1. Pointer and mask table.

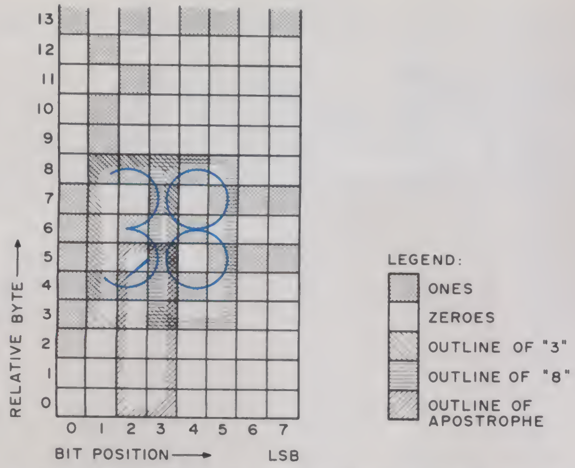


Fig. 3. Detail from dots table showing overlap of "3," "8" and apostrophe.

were to do a text editor using this algorithm, I would arrange for pushing the character onto the next line.

A better approach, if you assume reasonable-length text lines (on a screen more than 64 pixels wide), might be to let the screen be a "window" on the file, which may be slid around both in the vertical and horizontal directions. Either solution requires some careful forethought.

Another little space-saver I put in also relates to the line breaks. Each character cell as shown in Fig. 4 is to be augmented by one column of blank pixels on its right for character separation. But if a character ends exactly on the rightmost pixel of the screen, the line break will provide that necessary separation. In this case there is not much point in inserting the blank at the beginning of the next line, so I don't. It is not much of a savings, but it helps.

The result of all these machinations is an average of 18 characters per line in Tiny *BASIC* (by actual count). I would expect slightly better than that for ordinary (caps and lowercase) text.

Program Logic

The program to do all this is designed as a subroutine. It is called with the character to be displayed in the accumulator. The subroutine analyzes the character to determine whether it is a control character or not, then takes appropriate action.

The subroutine cleaves logically into two parts with a little housekeeping at the entry and exit. The two parts process the control functions (carriage return, line feed, back space, form feed and cursor blink) and the displayable characters, respectively. Much of the housekeeping is specific to the 1802 CPU architecture and register conventions, and I will get into those details in the second half of this article. For now I will assume that the necessary pointers are readily available (in registers or directly accessible RAM).

Other housekeeping functions include discriminating between the various control characters to be processed and ignoring the rest (i.e., return without taking any action) and separating out the text characters.

On exit, some of the control functions and all displayed characters are capable of running off the bottom of the screen. In this case the display must be scrolled up and the bottom line blanked.

The carriage-return and line-feed control functions simply adjust the position of the cursor. For use with Tiny BASIC it is desirable to make these functions emulate their teletypewriter equivalents.

In other words, a carriage return returns the cursor position to the left margin on the current line; a line feed moves the cursor down six pixels (one text line) but does not change its horizontal position. For text-processing applications, the line feed might be ignored and the carriage return might effect the composition of the two functions.

The form-feed control function is used to clear the screen and place the cursor in the upper left corner. Since the character to be displayed is built on the screen above the cursor, this requires that the form-feed command be followed by a line feed. Otherwise, the character is generated outside the buffer space.

In my program the form-feed service routine falls through the carriage-return routines; so if the composite action carriage return is implemented (by deleting a branch), the effective line feed will be included with the form feed as well.

Two control codes are used to blink the pixel that the cursor points to. This may not be as elegant as a whole underscore or a full character cell, but it was easy. I also did not have to decide how wide to make it: It is one pixel wide. One control (I use character 00 = NUL) blanks that pixel; the other (using code 01 = SOH) turns it on. These are also handy for plotting.

The back-space control function is the most complicated. If the back-space stack is empty there is no information about how far to back up and nothing more is done. The next concern is whether the bit pointer (i.e., the low three bits of the horizontal cursor position) saved in the stack represents the same byte as the present position of the cursor.

If not, then two clearing steps are required: one for the current column of six bytes and one for the column to the left. On the other hand, if both pointers are in the same word, only one set of six bytes needs clearing.

The back-space command notes the new cursor position, which is the same as the cursor position saved in the stack, and clears everything in its byte to the right. If the character was split across two bytes, the second (right hand) byte would be completely cleared on the assumption that it had no other text than the piece of character being erased. This latter part is done first, since the cursor already points to this block of six bytes.

Clearing the partial byte is a little tricky. First a mask is created with ones in its more significant bits and zeros in its less significant bits. The zeros correspond exactly to the pixels of the character to be erased. Each of the six bytes in turn is ANDed with this mask.

REFERENCE	DATA							
0	00	00	80	80	E8	97	A8	97
8	68	40	40	20	40	AD	B6	AD
16	44	04	20	56	DD	57	20	00
24	F4	AA	A9	AA	F4	00	39	E9
32	AB	AD	29	00	55	FA	54	F8
40	54	00	C0	24	4A	91	60	0A
48	55	75	51	51	20	14	EC	86
56	4C	27	E4	00	15	15	BE	55
64	B6	00	01	C2	3C	D2	91	F0
72	10	02	57	2A	70	A0	58	00
80	DE	68	A4	62	1C	06	23	55
88	15	18	10	51	04	22	A1	44
96	00	AF	AC	D6	AC	AF	00	42
104	42	5F	52	F9	10	00	46	45
112	56	6D	46	00	94	94	56	2D
120	EE	00	68	94	B4	B4	54	00
128	DF	A5	DF	80	80	35	55	3A
136	00	00	56	AA	AB	00	00	00

Table 2. Dots table.


Been waiting for the price of owning a computer to come down?
The time has come.
For less than $1.25 an hour you can own the DSC-2 from DIGITAL MICROSYSTEMS.
This effective business tool is from a company that has delivered over 2,000 computer and disk systems in the last two years alone.
The DSC-2 comes complete with a 120 cps printer, full-featured video display terminal, full-sized, double density floppy disk drives and a fast and expandable CPU. Enough power, speed and storage for the small business or professional office.
The remarkable capabilities of the DSC-2 can grow with your business or practice. Additional terminals, faster printers and up to 27.4 million characters of disk storage can be added.
The value of a small business computer is the job it can do for you. The DSC-2 Computer System can do your job and start doing it today. How? With well-designed application software packages that are fast to learn and easy to use. Packages like MEDIC — complete medical accounting for physicians or General Accounting — with Accounts Receivable, Accounts Payable, Payroll and General Ledger.
The DSC-2 costs less — $7,000 to $10,000 less — than other computer systems advertising the same capabilities. No other computer can give you so much for so little.
The small business computer you've been wanting is here at the price you've been waiting for.
$9,950.
If you've waited long enough, call or write us:
Digital Microsystems
4448 Piedmont Ave. • Oakland, CA 94611
(415) 658-8650
OEM and dealership discounts available.

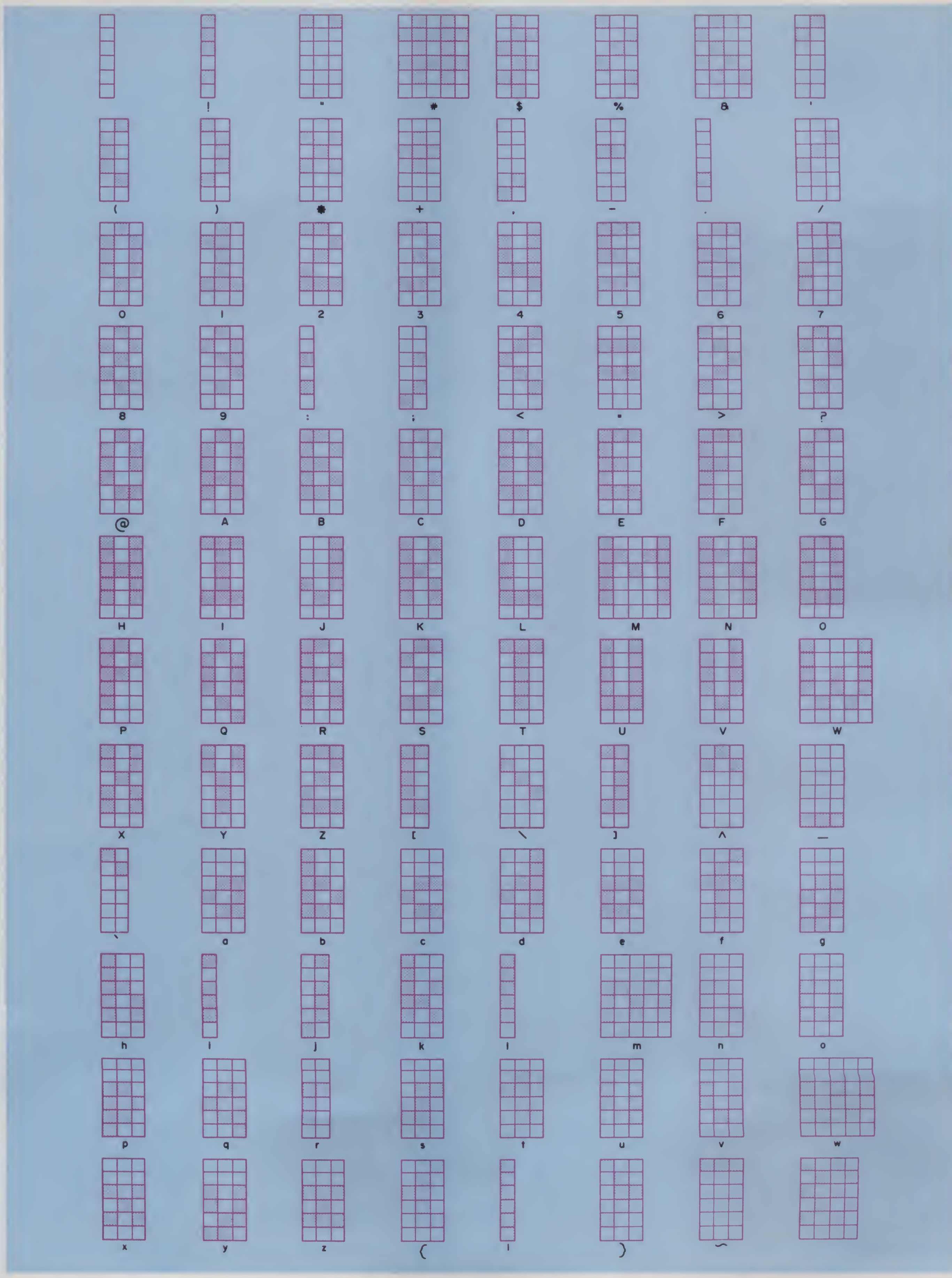


Fig. 4. Dot matrix pattern for 96 characters.

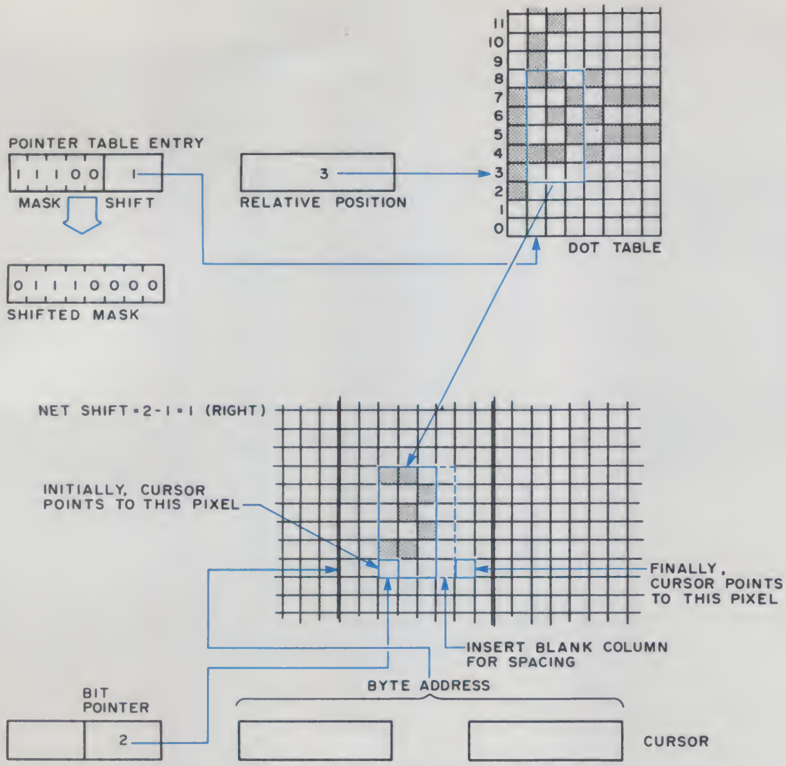


Fig. 5. Transfer of character to display buffer.

Because I already had a co-routine to alternately shift a predetermined number of places and then to bump the byte pointer up one pixel row, I used this routine here to re-create the mask for each byte. On an 8080 or Z-80 I suppose I might have coded it in line, using the double add instruction to bump the pointer up and creating the mask only once.

There is one hazard to watch for in the back spacing across a pair of bytes. That is the possibility that you are back spacing over a character split between lines. In such a case the vertical cursor position must be adjusted also. This is tricky only insofar as you fail to notice that (at least in the 1861 display buffer) the next byte to the left is one pixel row up; five more pixel rows must be subtracted to get the previous text line.

Transfer of Characters

Processing the display characters involves a few steps: The ASCII code is doubled and used to index into the pointer table. The second byte of the selected table entry is then used to index into the dot table, which points to the first (or bottom) row of the character.

Meanwhile, the bit mask is shifted to the right by the amount indicated in the low three bits of the mask-position byte of the position table entry. This positions it to select out the dots for the desired character and to zero the excess bits. Then the difference is computed between the table position of the character and the cursor position where it is to go. This difference represents how much it needs shifting and which direction.

If the character in the dot table occupies the same relative position in the byte as the place where it is to go in the display, no shift is necessary. Otherwise, it may need shifting up to seven bit positions either left or right. The signed shift amount is saved so that it may be applied to each of the six bytes of the character. The process is illustrated in Fig. 5.

As I mentioned in connection with the back-space command, I use a special co-routine to alternately shift the byte and bump the pointer. A co-routine is like a subroutine, but you do not specify a starting address (except initially). When you do a co-call, the co-routine is started up wherever it left off. When it finishes its task, it does another co-call to return to where the "main" routine left off.

In this particular situation the only advantage of coding the functions up as a co-routine was to save time and space. In the 1802 a co-call costs only one byte of code and one fetch-execute cycle. In the 8080 the same result could be easily achieved with RST (restart) instructions or even with ordinary, plain-vanilla subroutine call instructions (JSR or BSR in the 6800 or 6502). It is not often that a program really needs the peculiar capability of co-routines.

Anyway, the first co-call shifts the byte the correct amount and direction. On return it is stored into the buffer using a logical OR to preserve previous contents. The second co-call then bumps the vertical part of the cursor position, so that it points to the next byte up. The sixth time through, the cursor pointer is restored to its original position and the carry is cleared to indicate that the loop is complete.

Now it is possible that what has been stored was not the whole character. It might have to be split across bytes, and

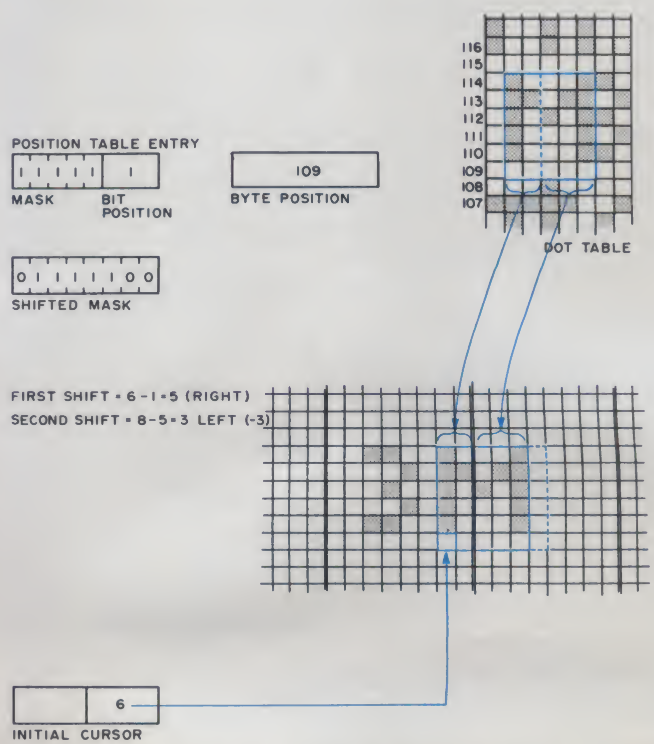


Fig. 6. Transfer of character with byte split.

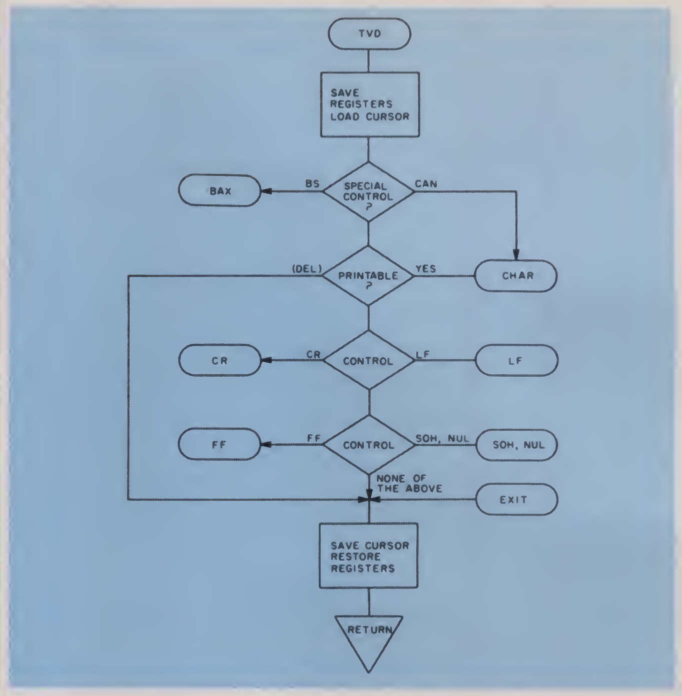


Fig. 7. Overall flow.

this loop only stored the left part of the character into the display buffer. The only way to tell for sure is to shift the mask the same number of bits, then see if any bits come off the end.

You cannot depend on any bits coming off the end when the character itself is shifted because the particular byte you look at may have zeros shifted off. But after the mask is shifted, if there are any bits lost off the end, at least one of them will be in the carry. This is because we require that there be no holes in the mask.

If the character is indeed split, the dots pointer, which we had been incrementing along as we fetched the six bytes of this character, must be backed up. In the 6502 you might use the Post-indirect Indexed addressing mode to access the dot table. Backing up the pointer then becomes as easy as clearing Y. The byte pointer part of the cursor must be advanced to the next byte, and if

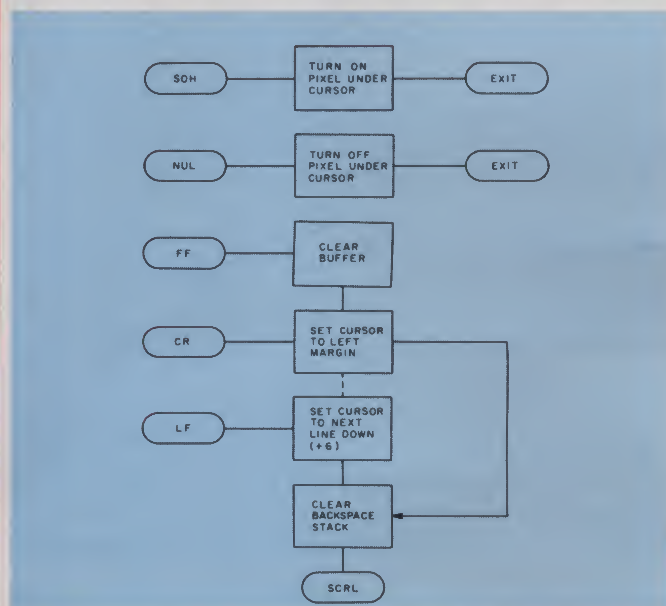


Fig. 8. Control codes.

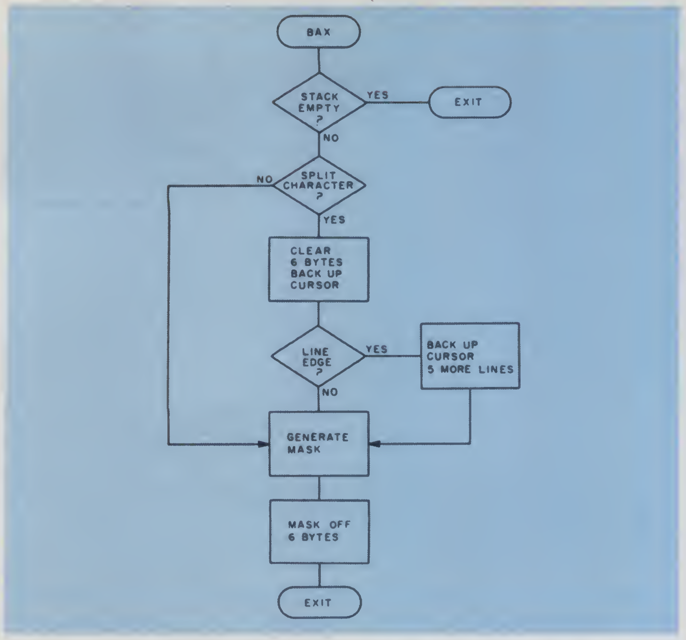


Fig. 10. Back space.

that crosses a line boundary, the vertical part of the cursor pointer must also be advanced (down five pixels).

We know that the shift counter was positive (to the right) in order to result in a character split. This is obvious because a character in the dot table is necessarily wholly contained in its byte, and any shift to the left must still leave it wholly within the byte (otherwise, it would overlap with the previous character to the left, which would be a bug).

To get the rest of the character it is necessary to shift it left by eight's complement of the previous right shift. This is illustrated in Fig. 6 with an "M."

There are two more bits available in the first byte (the bit pointer of the cursor is 6) so the first cycle shifts right 5 (the difference between 6, the cursor position, and 1, the position of the "M" in the dot table). The mask, 01111100, loses three bits when shifted right, so the second byte processing happens. The eight's complement of 5 is 3, so we need to shift left 3 bits. This is equivalent to shifting right 5 then left 8 (assuming no bits are lost in the process).

After the second time through the loop, the mask shifting step would again return a "bits are lost" response, except that the left-shift routine anticipates the problem and always clears the carry. This prevents an infinite loop on split characters.

Finally, after the whole character has been stored into the buffer, the (shifted) mask is shifted to the right one bit at a time while counting the shifts to find the right edge of the stored character. One is added to the negative (eight's complement again) of this count, and we look again to be sure the re-

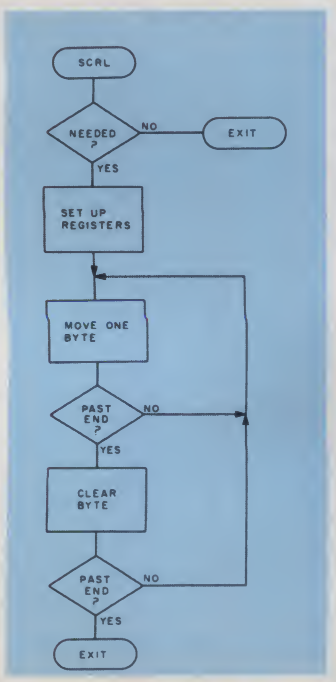


Fig. 9. Scrolling.

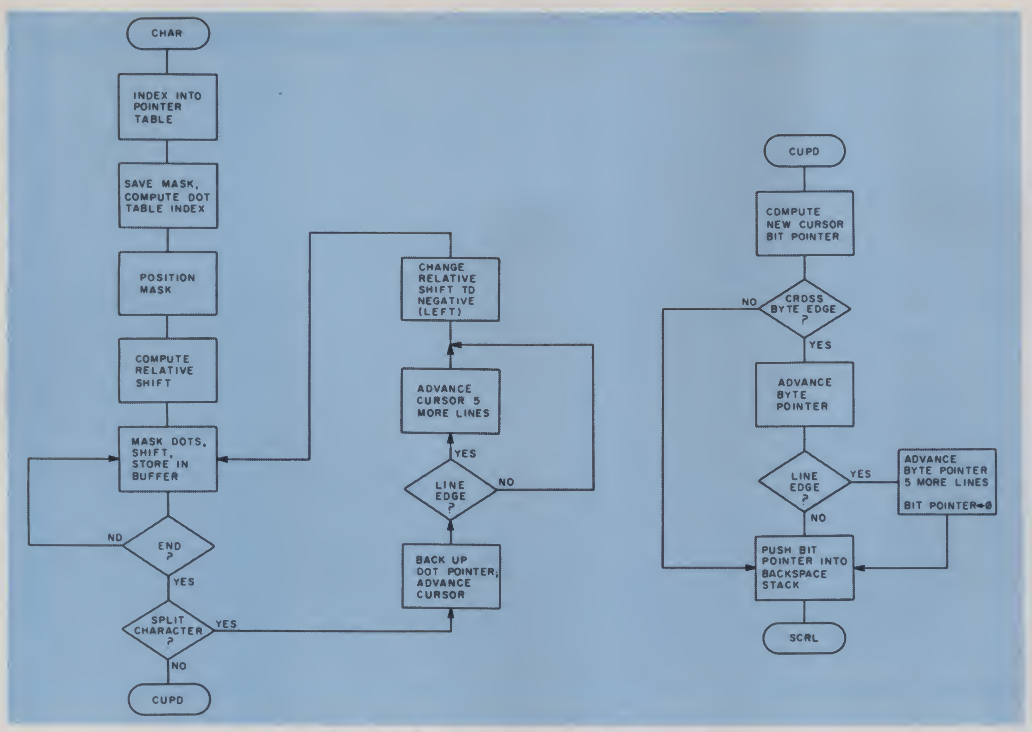


Fig. 11. Display character generation.

sult is not zero (or 8, which is the same thing in eight's complement) or one. If so, the byte-pointer part of the cursor still needs incrementing.

One last step is required before exiting to the scrolling test. The new bit pointer must be pushed into the backup stack. It is not critical whether the pointer is pushed at the beginning of the routine or at the end (as I do it), but the backspace command processor must know which.

In the shift subroutine (co-routine in my code) I not only shift the byte by the requisite amount, but I also check for the cursor lying outside the display buffer area. If the cursor is outside the display buffer, the bits are returned zero so that the OR/store operation will not modify memory.

Since the shift count and its sign occupy only four of the eight bits in a byte, I use the other four bits to count the number of rows. When the row count reaches six, the byte pointer must be bumped down five rows instead of up one row.

A Challenge

Now you have a notion about the general logic of the driver part of this program. Figs. 7-13 contain flowcharts to aid your understanding. The actual program also has a keyboard input routine that blinks the cursor and the hardware-specific interrupt service routine for the 1861. These are discussed in more detail in the second half of this article. Those of you without specific interest in the world's most powerful 8-bit microprocessor can jump ship at this port.

Before you go, let me put a challenge to you: I am convinced that the 1802 both runs faster (when normalized for memory cycle time) and uses less memory than any other 8-bitter. If you disagree with me, write (and debug!) a functionally equivalent program for your favorite (8-bit) CPU.

I have not spent any significant effort optimizing either parameter (time or space) in my program, but I think it would be worth $50 out of my own pocket to the first person who can show a 10 percent improvement in program efficiency as measured by the product of the number of bytes of code times the average number of memory cycle equivalents (as a measure of speed: three 8080 clock cycles is one memory cycle; in the 6800 and 6502 memory cycles at the clock rate; in the 1802 it is eight clock cycles).

The part of the program to be measured is that which I have described so far, consisting of a callable subroutine to encode uppercase and lowercase ASCII into a memory buffer (64 pixels wide by 42 high), with its required subroutines and tables. You do not have to use my program logic, but you do have to produce the same results (including control characters).

I will personally reply to any serious entries mailed to me, but I reserve the right to require evidence that the program runs correctly before awarding the prize. For reference, the 1802 program you are trying to beat is 826 bytes (including tables), and average execution time is 1375 memory cycles (formula: $N = H + D \times 95\% + C \times 4\% + B \times 1\% + S \times 5\%$). The byte-cycle product is therefore 1.13 million. Anything under a million is a clear winner.

We'll take a look at my code for the 1802 next month in Part 2 of this article. ■

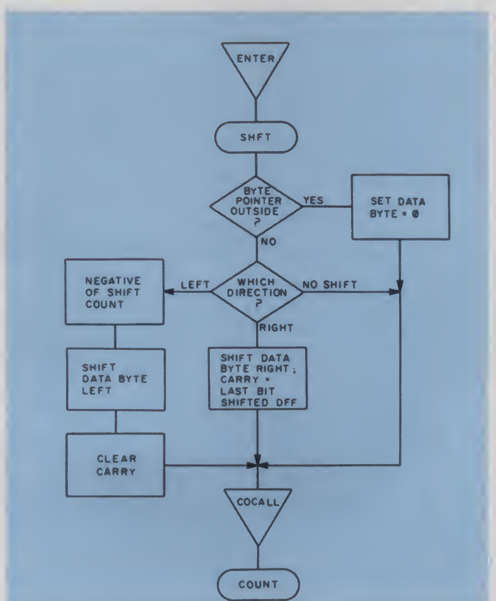


Fig. 12. Shift co-routine.

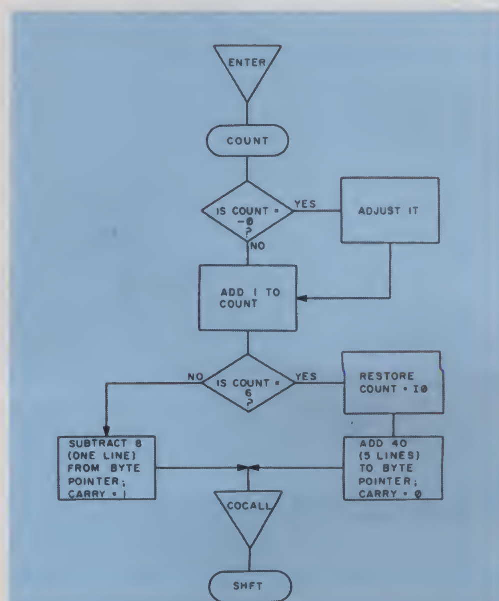


Fig. 13. Count co-routine.

Specifications:
S-100 compatible. MFM encoding, 35 tracks with ten 512-byte sectors per track. 179,200 bytes on double density SA-400 and North Star BASIC, DOS, and Monitor included.

For further information, write for full color catalog or contact your local computer store.

New from North Star
Double Density Performance at Single Density Prices

The new HORIZON computer and Micro Disk System now record in double density! That means each new Shugart SA-400 minifloppy disk drive accesses 180K bytes of on-line information. All double density HORIZON computers and Micro Disk Systems have a redesigned controller which allows the use of quadruple capacity disk drives as they become available in early 1979. A three-drive North Star System with quadruple capacity disk drives will access over a megabyte of on-line information. But, best of all there's no price increase for double density models.

North Star BASIC and DOS have been upgraded to accommodate the increased capacity and yet run existing programs with little or no change. The new disk system also supports single density, so existing single density diskettes can still be used. Single density SA-400 drives previously purchased with North Star systems can also be used.

Pricing

HORIZON with one double density SA-400 minifloppy (180K bytes), 16K RAM, Z80A processor and serial I/O port: $1599 kit, $1899 assembled.

MICRO DISK SYSTEM with one double density SA-400 minifloppy, controller board and power regulation: $699 kit, $799 assembled. (Cabinet and power supply $39 extra each.)

NORTH STAR ★ COMPUTERS
2547 Ninth Street
Berkeley, California 94710
(415) 549-0858

N9

The Apple Speaks . . . *Softly*

An inexpensive talking Apple . . . no, you're not in Wonderland.

Robert J. Bishop
1143 W. Badillo #E
Covina CA 91722

The Apple II computer is really a remarkable machine. It comes complete and ready to use; all you add is your own TV set or video monitor. The Apple even has a built-in speaker for "beeping" at you when you make a mistake in your program. But since the speaker is software accessible, it can be used for more than just producing simple beeps. In fact, there are programs around that use the speaker for everything from playing music to generating phaser sounds when you shoot at Klingons.

Also built into the standard Apple II is all the hardware required for saving and loading programs to and from cassette tape. The monitor firmware includes routines that read and write a series of square waves that contain the bits to be transferred between the computer's memory and your tape recorder.

Then one day it hit me. If the Apple can read a series of square waves from the tape, might it not be possible for it to read other waveforms, too? Of course, since the data seen coming in through the cassette port can only be binary (signal high or signal low), any complicated waveform will still appear to be a series of square waves (see Fig. 1). However, there might still be enough of the fundamental information retained after digitization to allow for a crude reconstruction of the input signal when played back through the built-in speaker. It was certainly worth a try!

To test the concept I wrote the short program that accompanies this article. All it does is sample the cassette input port (location $C060) until it detects a change in the input signal level (bit 7). When the input signal makes a transition, the program toggles the speaker (location $C030) to reflect the change.

I could hardly wait to try it out. I grabbed a blank tape, stuck it into my cassette recorder and said something like, "Testing, one, two, three." I rewound the tape and then played it into the computer.

Much to my surprise it actually worked! I could hear my voice coming through the Apple's speaker! Of course, the fidelity left a lot to be desired; it reminded me of the way that a tiny transistor radio sounds when you turn the volume up too high. But it was, nevertheless, surprisingly intelligible.

As a next step, it was fairly easy to modify the program to store the digitized results in memory and then *later* play it through the speaker, thus creating a kind of "digital tape recorder."

What other applications are there for this technique? Well, like computing in general, what you can do is limited only by your imagination. Some ideas include:

- A program that pronounces ASCII characters as they are printed on the screen.
- Speech recognition experiments.
- A kaleidoscopic display whose motion is controlled by input music.
- Frequency analyzer.
- All kinds of neat what-cha-mi-call-its.

In closing, I might just mention that there are several companies selling voice recognition and synthesis *hardware* for hundreds of bucks. (Some of these are even made for the Apple!) Well, now you can do it yourself . . . in *software.*■

```
              :APPLE-II VOICE
              :DIGITIZER DEMO
              :
              TEMP .DL 0000
              SPKR .DL C030
              TAPE .DL C060
              :
0800 AD60C0   MAIN LDA TAPE
0803 8500          STA *TEMP
0805 AD60C0   LOOP LDA TAPE
0808 4500          EOR *TEMP
080A 10F9          BPL LOOP
080C 8D30C0        STA SPKR
080F 4C0008        JMP MAIN
              :
                   .EN

SYMBOL TABLE
TEMP    0000
SPKR    C030
TAPE    C060
MAIN    0800
LOOP    0805
```

Program listing.

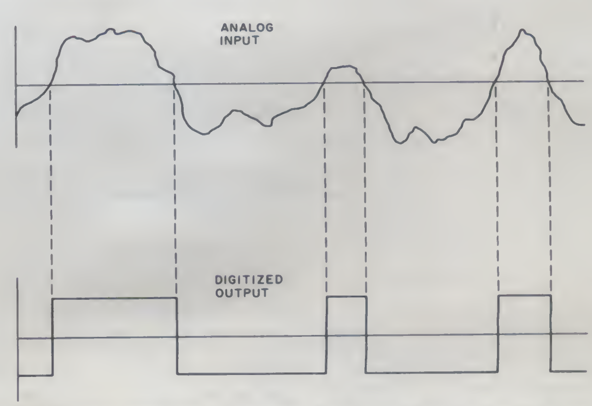


Fig. 1. Correspondence between the input and output signals of the digitizer.

Super Mastermind

Super Mastermind is a challenging two-player game of logic, deduction and intuition.

Tom Cardoso
258 Taylor Ave.
Glen Ellyn IL 60137

There is a myriad of two-player games converted for use on computers (probably because you can be assured a computer won't complain or cheat and will tirelessly play game after game for as long as the player can hold up). Super Mastermind is but another of these games—but oh, what a challenge!

Mastermind is an enticing game of deduction where a five-color code is hidden by the code-maker—in this case, the computer. The object of the game is to break the code—to identify the exact colors chosen for the code and determine their precise order. The computer chooses the five-unit code at random from up to nine colors and hides them in a row. As the code-breaker, you have 12 attempts to establish which five (of the nine) were chosen by the computer and how they are placed in the row. After each five-digit guess, the computer gives you a white marker for each correct color and a black marker if a correct color *and* its precise location in the row have been properly identified. By logically analyzing the markers after each guess, you will eventually be able to identify the code.

This program, written for SOL BASIC5, is intended for use with a SOL-20 computer with SOLOS personality module. The simplified nature of BASIC5, however, should permit easy conversion to other forms of BASIC. The program will run in 14K of memory, including about 6.5K allocated for BASIC5.

Because SOL BASIC5 does not have string-handling capabilities, it is necessary to use numbers as replacements for colors. Consequently, the computer generates the color code through a series of five one-digit numbers (from 1 to 9). The object of the game remains the same, however—to break the code. This program plays identical to the store-bought table version that uses colored marbles or chips. In this case, the numbers are simply interpreted as colors.

A unique feature of this program, which can be utilized by other programs requiring random-number generation, is its own random-number generator. Why not use the internal generator residing in BASIC5? Well, you could, except that every time BASIC is initialized it causes the same sequence of random numbers to be generated. Soon a player would tire of knowing that the first and all subsequent games have the same sequence of numbers. The codes generated will follow the same sequence every time the game is loaded into the computer. With this generator, the player has the option of changing the game sequence simply by inputting any seven-digit decimal. This seed is then used to generate the random numbers in determining the code. As long as the seed is different for each series of games, the sequence generated will be different. That way you will not be able to guess in advance what the code is.

Another feature of this program is the ability to choose the number of colors used to make the code. A beginning

Program listing.

```
LIST
1     REM ********** MMIND **********
2     REM SUPER MASTERMIND *** Written for SOL BASIC5 *** December, 1977
3     REM Written by Tom Cardoso  Glen Ellyn, Illinois
4     REM "&K" STATEMENTS USED TO CLEAR SCREEN
7     PRINT "&K": PRINT : PRINT : PRINT : M=0: N=1: S9=.5284163
10    PRINT TAB(10),"This is the game of SUPER MASTERMIND."
12    PRINT : PRINT TAB(10),: INPUT "Do you want the rules?  0=NO, 1=YES  ",P
15    IF P=1 THEN 39
20    IF P>1 THEN 10
30    IF P=0 THEN PRINT : GOTO 87
39    PRINT "&K": PRINT TAB(7),"^^^^^^^ WELCOME TO SUPER MASTERMIND ^^^^^^^": PRINT
42    PRINT "In this game of logic and deduction, your intuitive genius"
45    PRINT "will be tested by having you try to determine the"
50    PRINT "exact sequence of 5 one-digit numbers (colors) chosen"
55    PRINT "at random from up to 9 possibilities (numbers used will be"
60    PRINT "1-9).  Duplicate numbers are possible.  In fact, all"
65    PRINT "five selections can be the same.  After each guess, you"
70    PRINT "will be told how many of your guesses correspond to the"
75    PRINT "unseen random selection.  A 'BLACK' marker is given"
80    PRINT "if both 'color' and row are correct.  A 'WHITE' marker"
82    PRINT "is given if only 'color' is correct.  Input your guess"
83    PRINT "with commas following each digit, then hit 'return'."
84    PRINT : INPUT "TYPE '1' IF YOU ARE READY... ",P: IF P<>1 THEN 84
87    PRINT "&K": PRINT : PRINT : PRINT "In order to prevent a series of games from playing"
88    PRINT "identically to the last series, you may wish to change"
89    PRINT "the sequence by entering a random, uneven seven digit decimal."
90    INPUT "If no change in the sequence is desired, enter '0'.  ",S8
92    IF S8>1 THEN PRINT : INPUT "Input a DECIMAL, please...  ",S8: IF S8>1 THEN 92
93    IF S8>0 THEN S9=S8
94    PRINT "&K": PRINT : PRINT : PRINT "There are  3125 combinations possible with 5 'colors'."
95    PRINT "There are  7776 combinations possible with 6 'colors'."
96    PRINT "There are 16807 combinations possible with 7 'colors'."
```

```
97   PRINT "There are 32768 combinations possible with 8 `colors'."
98   PRINT "There are 59049 combinations possible with 9 `colors'."
99   PRINT : INPUT "How many `colors' do you wish to play with? (5 to 9)  ",J9
100  IF J9<5 THEN 94
101  IF J9>9 THEN 94
102  PRINT : PRINT "Your code will consist of `colors' from 1 to";J9
103  PRINT : PRINT "Let me display the game board...": FOR X9=1 TO 2000: NEXT
105  REM CLEAR GAME BOARD VARIABLES
106  A=0: B=0: C=0: D=0: E=0: F=0: G=0: H=0: I=0: J=0: K=0: L=0: A1=0: B1=0
107  C1=0: D1=0: E1=0: K1=0: L1=0: M1=0: N1=0: O1=0: P1=0: Q1=0: R1=0: S1=0
108  A2=0: B2=0: C2=0: D2=0: E2=0: K2=0: L2=0: M2=0: N2=0: O2=0: P2=0: Q2=0
109  R2=0: S2=0: A3=0: B3=0: C3=0: D3=0: E3=0: K3=0: L3=0: M3=0: N3=0: O3=0
110  P3=0: Q3=0: R3=0: S3=0: A4=0: B4=0: C4=0: D4=0: E4=0: K4=0: L4=0: M4=0
111  N4=0: O4=0: P4=0: Q4=0: R4=0: S4=0: A5=0: B5=0: C5=0: D5=0: E5=0: K5=0
112  L5=0: M5=0: N5=0: O5=0: P5=0: Q5=0: R5=0: S5=0: A6=0: B6=0: C6=0: D6=0
113  E6=0: K6=0: L6=0: M6=0: N6=0: O6=0: P6=0: Q6=0: R6=0: S6=0
114  S=0: T=0: U=0: V=0: W=0: K9=J9+1: GOTO 140
119  REM RANDOM NUMBER GENERATOR SUBROUTINE
120  W9=S9*997: W8=W9-INT(W9): S9=W8: RETURN
139  REM GENERATE CODE
140  GOSUB 120: A=INT(W8*K9): IF A=0 THEN 140
142  GOSUB 120: B=INT(W8*K9): IF B=0 THEN 142
144  GOSUB 120: C=INT(W8*K9): IF C=0 THEN 144
146  GOSUB 120: D=INT(W8*K9): IF D=0 THEN 146
148  GOSUB 120: E=INT(W8*K9): IF E=0 THEN 148
150  GOTO 229
220  INPUT "GUESS ?...  ",K7,L7,M7,N7,O7
221  IF K7>9 THEN 227
222  IF L7>9 THEN 227
223  IF M7>9 THEN 227
224  IF N7>9 THEN 227
225  IF O7>9 THEN 227
226  RETURN
227  PRINT "DON'T FORGET THE COMMAS !!": GOTO 220
229  GOSUB 2000: K=0: L=0: GOSUB 220: A1=K7: B1=L7: C1=M7: D1=N7: E1=O7
230  M=M+1: F=A1: G=B1: H=C1: I=D1: J=E1: GOSUB 3000: P1=K: Q1=L
265  IF K=5 THEN 3500
280  GOSUB 2000: K=0: L=0: GOSUB 220: K1=K7: L1=L7: M1=M7: N1=N7: O1=O7
290  M=M+1: F=K1: G=L1: H=M1: I=N1: J=O1: GOSUB 3000: R1=K: S1=L
295  IF K=5 THEN 3500
340  GOSUB 2000: K=0: L=0: GOSUB 220: A2=K7: B2=L7: C2=M7: D2=N7: E2=O7
350  M=M+1: F=A2: G=B2: H=C2: I=D2: J=E2: GOSUB 3000: P2=K: Q2=L
355  IF K=5 THEN 3500
400  GOSUB 2000: K=0: L=0: GOSUB 220: K2=K7: L2=L7: M2=M7: N2=N7: O2=O7
410  M=M+1: F=K2: G=L2: H=M2: I=N2: J=O2: GOSUB 3000: R2=K: S2=L
420  IF K=5 THEN 3500
460  GOSUB 2000: K=0: L=0: GOSUB 220: A3=K7: B3=L7: C3=M7: D3=N7: E3=O7
470  M=M+1: F=A3: G=B3: H=C3: I=D3: J=E3: GOSUB 3000: P3=K: Q3=L
480  IF K=5 THEN 3500
500  GOSUB 2000: K=0: L=0: GOSUB 220: K3=K7: L3=L7: M3=M7: N3=N7: O3=O7
510  M=M+1: F=K3: G=L3: H=M3: I=N3: J=O3: GOSUB 3000: R3=K: S3=L
520  IF K=5 THEN 3500
580  GOSUB 2000: K=0: L=0: GOSUB 220: A4=K7: B4=L7: C4=M7: D4=N7: E4=O7
590  M=M+1: F=A4: G=B4: H=C4: I=D4: J=E4: GOSUB 3000: P4=K: Q4=L
600  IF K=5 THEN 3500
640  GOSUB 2000: K=0: L=0: GOSUB 220: K4=K7: L4=L7: M4=M7: N4=N7: O4=O7
650  M=M+1: F=K4: G=L4: H=M4: I=N4: J=O4: GOSUB 3000: R4=K: S4=L
660  IF K=5 THEN 3500
700  GOSUB 2000: K=0: L=0: GOSUB 220: A5=K7: B5=L7: C5=M7: D5=N7: E5=O7
710  M=M+1: F=A5: G=B5: H=C5: I=D5: J=E5: GOSUB 3000: P5=K: Q5=L
720  IF K=5 THEN 3500
760  GOSUB 2000: K=0: L=0: GOSUB 220: K5=K7: L5=L7: M5=M7: N5=N7: O5=O7
770  M=M+1: F=K5: G=L5: H=M5: I=N5: J=O5: GOSUB 3000: R5=K: S5=L
780  IF K=5 THEN 3500
800  GOSUB 2000: K=0: L=0: GOSUB 220: A6=K7: B6=L7: C6=M7: D6=N7: E6=O7
810  M=M+1: F=A6: G=B6: H=C6: I=D6: J=E6: GOSUB 3000: P6=K: Q6=L
820  IF K=5 THEN 3500
880  GOSUB 2000: K=0: L=0: GOSUB 220: K6=K7: L6=L7: M6=M7: N6=N7: O6=O7
890  M=M+1: F=K6: G=L6: H=M6: I=N6: J=O6: GOSUB 3000: R6=K: S6=L
895  IF K=5 THEN 3500
900  GOTO 4500
1999 REM GENERATE DISPLAY BOARD
2000 PRINT "&K": PRINT TAB(4),"*** Super Mastermind ***"
2010 PRINT "¦  ?  ¦  ?  ¦  ?  ¦  ?  ¦  ?  ¦¦¦¦",
2020 PRINT TAB(35),"# of Blacks    # of Whites"
2030 FOR P=1 TO 34: PRINT "~";: NEXT : PRINT
2040 PRINT "¦ ";A1;"  ¦ ";B1;"  ¦ ";C1;"  ¦ ";D1;"  ¦ ";E1;"  ¦¦¦¦",
2050 PRINT TAB(39),P1,TAB(54),Q1
2060 PRINT "¦ ";K1;"  ¦ ";L1;"  ¦ ";M1;"  ¦ ";N1;"  ¦ ";O1;"  ¦¦¦¦",
2070 PRINT TAB(39),R1,TAB(54),S1
2080 PRINT "¦ ";A2;"  ¦ ";B2;"  ¦ ";C2;"  ¦ ";D2;"  ¦ ";E2;"  ¦¦¦¦",
2090 PRINT TAB(39),P2,TAB(54),Q2
2100 PRINT "¦ ";K2;"  ¦ ";L2;"  ¦ ";M2;"  ¦ ";N2;"  ¦ ";O2;"  ¦¦¦¦",
2110 PRINT TAB(39),R2,TAB(54),S2
2120 PRINT "¦ ";A3;"  ¦ ";B3;"  ¦ ";C3;"  ¦ ";D3;"  ¦ ";E3;"  ¦¦¦¦",
2130 PRINT TAB(39),P3,TAB(54),Q3
2140 PRINT "¦ ";K3;"  ¦ ";L3;"  ¦ ";M3;"  ¦ ";N3;"  ¦ ";O3;"  ¦¦¦¦",
2150 PRINT TAB(39),R3,TAB(54),S3
2160 PRINT "¦ ";A4;"  ¦ ";B4;"  ¦ ";C4;"  ¦ ";D4;"  ¦ ";E4;"  ¦¦¦¦",
2170 PRINT TAB(39),P4,TAB(54),Q4
2180 PRINT "¦ ";K4;"  ¦ ";L4;"  ¦ ";M4;"  ¦ ";N4;"  ¦ ";O4;"  ¦¦¦¦",
2190 PRINT TAB(39),R4,TAB(54),S4
2200 PRINT "¦ ";A5;"  ¦ ";B5;"  ¦ ";C5;"  ¦ ";D5;"  ¦ ";E5;"  ¦¦¦¦",
2210 PRINT TAB(39),P5,TAB(54),Q5
2220 PRINT "¦ ";K5;"  ¦ ";L5;"  ¦ ";M5;"  ¦ ";N5;"  ¦ ";O5;"  ¦¦¦¦",
2230 PRINT TAB(39),R5,TAB(54),S5
2240 PRINT "¦ ";A6;"  ¦ ";B6;"  ¦ ";C6;"  ¦ ";D6;"  ¦ ";E6;"  ¦¦¦¦",
2250 PRINT TAB(39),P6,TAB(54),Q6
2260 PRINT "¦ ";K6;"  ¦ ";L6;"  ¦ ";M6;"  ¦ ";N6;"  ¦ ";O6;"  ¦¦¦¦",
2270 PRINT TAB(39),R6,TAB(54),S6
2280 RETURN
2999 REM GUESS ANALYSIS FOR `BLACK' OR `WHITE'
3000 S=A: T=B: U=C: V=D: W=E
3110 IF F=S THEN K=K+1: S=0: F=10
3120 IF G=T THEN K=K+1: T=0: G=10
3130 IF H=U THEN K=K+1: U=0: H=10
3140 IF I=V THEN K=K+1: V=0: I=10
3150 IF J=W THEN K=K+1: W=0: J=10
3160 IF F=T THEN L=L+1: T=0: F=10
3170 IF F=U THEN L=L+1: U=0: F=10
3180 IF F=V THEN L=L+1: V=0: F=10
3190 IF F=W THEN L=L+1: W=0: F=10
3200 IF G=S THEN L=L+1: S=0: G=10
3210 IF G=U THEN L=L+1: U=0: G=10
3220 IF G=V THEN L=L+1: V=0: G=10
3230 IF G=W THEN L=L+1: W=0: G=10
3240 IF H=S THEN L=L+1: S=0: H=10
3250 IF H=T THEN L=L+1: T=0: H=10
3260 IF H=V THEN L=L+1: V=0: H=10
3270 IF H=W THEN L=L+1: W=0: H=10
3280 IF I=S THEN L=L+1: S=0: I=10
3290 IF I=T THEN L=L+1: T=0: I=10
3300 IF I=U THEN L=L+1: U=0: I=10
3310 IF I=W THEN L=L+1: W=0: I=10
3320 IF J=S THEN L=L+1: S=0: J=10
3330 IF J=T THEN L=L+1: T=0: J=10
3340 IF J=U THEN L=L+1: U=0: J=10
3350 IF J=V THEN L=L+1: V=0: J=10
3360 RETURN
3499 REM GENERATE WINNER BOARD WITH CODE DISPLAY
3500 PRINT "&K": PRINT TAB(4),"*** Super Mastermind ***"
3510 PRINT "¦ ";A;"  ¦ ";B;"  ¦ ";C;"  ¦ ";D;"  ¦ ";E;"  ¦¦¦¦",
3520 PRINT TAB(35),"Congratulations!!  You"
3530 FOR P=1 TO 34: PRINT "~";: NEXT
3540 PRINT TAB(35),"must be a GENIUS!!": GOSUB 2040
3550 PRINT "Your average for ";N;"  games is ";%Z1%;M/N;"  guesses to win!";%%
3560 INPUT "Do you care to play another game?  0=NO, 1=YES  ",P
3570 IF P=1 THEN N=N+1: GOTO 94
3580 IF P>1 THEN 3560
3585 REM RESET GAME AVERAGE FOR NEW PLAYER
3590 PRINT "&K": PRINT : PRINT : INPUT "Does someone else want to play?  0=NO, 1=YES  ",P
3600 IF P=1 THEN 7
3610 IF P>1 THEN 3590
3620 PRINT : PRINT : PRINT : PRINT "THANKS A LOT FOR THE GAME...HOPE YOU HAD FUN": END
4499 REM GENERATE LOSER BOARD WITH CODE DISPLAY
4500 PRINT "&K": PRINT TAB(4),"*** Super Mastermind ***"
4505 PRINT "¦ ";A;"  ¦ ";B;"  ¦ ";C;"  ¦ ";D;"  ¦ ";E;"  ¦¦¦¦",
4510 PRINT TAB(35),"Sorry, you couldn't"
4520 FOR P=1 TO 34: PRINT "~";: NEXT
4530 PRINT TAB(35),"quite get it.  Try again.": GOSUB 2040
4535 INPUT "Do you care to play another game?  0=NO, 1=YES  ",P
4540 IF P=1 THEN N=1: M=0: GOTO 94
4545 IF P>1 THEN 4535
4550 GOTO 3590
4560 END
```

```
          This is the game of SUPER MASTERMIND.

          Do you want the rules?  0=NO, 1=YES  0


In order to prevent a series of games from playing
identically to the last series, you may wish to change
the sequence by entering a random, uneven seven digit decimal.
If no change in the sequence is desired, enter 0.  0

There are  3125 combinations possible with 5 'colors'.
There are  7776 combinations possible with 6 'colors'.
There are 16807 combinations possible with 7 'colors'.
There are 32768 combinations possible with 8 'colors'.
There are 59049 combinations possible with 9 'colors'.

How many 'colors' do you wish to play with? (5 to 9)  9

Your code will consist of 'colors' from 1 to 9

Let me generate the game...

         SUPER MASTERMIND
! ? ! ? ! ? ! ? ! ? !!!! # OF BLACKS    # OF WHITES
-------------------------------------
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
GUESS?  2,3,8,7,9

         SUPER MASTERMIND
! ? ! ? ! ? ! ? ! ? !!!! # OF BLACKS    # OF WHITES
-------------------------------------
! 2 ! 3 ! 8 ! 7 ! 9 !!!!     1              1
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
GUESS?  2,5,8,9,7

         SUPER MASTERMIND
! ? ! ? ! ? ! ? ! ? !!!! # OF BLACKS    # OF WHITES
-------------------------------------
! 2 ! 3 ! 8 ! 7 ! 9 !!!!     1              1
! 2 ! 5 ! 8 ! 9 ! 7 !!!!     2              1
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
GUESS?  8,1,5,9,9

         SUPER MASTERMIND
! ? ! ? ! ? ! ? ! ? !!!! # OF BLACKS    # OF WHITES
-------------------------------------
! 2 ! 3 ! 8 ! 7 ! 9 !!!!     1              1
! 2 ! 5 ! 8 ! 9 ! 7 !!!!     2              1
! 8 ! 1 ! 5 ! 9 ! 9 !!!!     3              2
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
GUESS?  8,5,1,9,9

         SUPER MASTERMIND
! 8 ! 5 ! 1 ! 9 ! 9 !!!! CONGRATULATIONS!!  YOU
-------------------------------------- MUST BE A GENIUS!!
! 2 ! 3 ! 8 ! 7 ! 9 !!!!     1              1
! 2 ! 5 ! 8 ! 9 ! 7 !!!!     2              1
! 8 ! 1 ! 5 ! 9 ! 9 !!!!     3              2
! 8 ! 5 ! 1 ! 9 ! 9 !!!!     5              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
! 0 ! 0 ! 0 ! 0 ! 0 !!!!     0              0
YOUR AVERAGE FOR  1  GAMES IS  4.0  GUESSES TO WIN!
DO YOU CARE TO PLAY ANOTHER GAME?  0=NO, 1=YES  0

DOES SOMEONE ELSE WANT TO PLAY?  0=NO, 1=YES  0
```

Sample run.

player may want to start with just five or six different combinations. (The number of possible combinations is significantly lower with five than when all nine are chosen for play.) There are 3125 possible combinations when five colors are used, but 59,049 combinations when nine are used! Always remember that in any code a color can be chosen more than once. In fact, all five choices in the code can be the same color!

For those who do not wish to take the time to enter this program manually, CUTS format cassette tapes are available from the author for $4.95, first-class postage paid. Comments and suggestions are always welcome. Good thinking and best of fun.■

✓ Reader Service—see page 179

INTRODUCING

DUAL DRIVE MINIFLOPPY FOR PET!

A FULL RANGE OF PERIPHERALS TO COMPLETELY SUPPORT YOUR PET + PROFESSIONAL SOFTWARE SUPPORT = A TOTALLY INTEGRATED PET* SYSTEM

DOMESTIC & OVERSEAS DEALER INQUIRIES INVITED ON DISK SYSTEM & MEMORY

THE DISKDRIVER, DOS, AND EXPANDAPET ARE ALL PRODUCTS OF COMPUTHINK

NOW THE DISKDRIVER DUAL DRIVE DISK SYSTEM CAN ADD TOTAL SYSTEM CAPABILITIES TO PET!

- ★ DUAL MINI FLOPPY DRIVE (5.5") WITH 100K PER DISK SIDE FOR TOTAL 200K ON LINE.
- ★ EXPANDABLE IN MARCH/APRIL 1979 TO 400K ON LINE VIA DOUBLE DENSITY FORMAT.
- ★ DISKMON-DISK OPERATING SYSTEM (DOS) IS RESIDENT IN PROM VIA DISK CONTROLLER BOARD THAT PLUGS DIRECTLY INTO THE INTERNAL EXPANDAPET MEMORY BOARD.*
- ★ DISKMON COMMANDS EXIST SIMULTANEOUSLY WITH THE BASIC LANGUAGE COMMANDS.
- ★ DISKMON ADDS 16 COMMANDS TO BASIC INCLUDING DISK DATA FILE COMMANDS.
- ★ DESIGNED FOR COMMERCIAL SPEED REQUIREMENTS — READS AN ENTIRE TRACK AT A TIME.
- ★ DISK DRIVES ARE AUTOMATICALLY TURNED OFF BETWEEN COMMANDS FOR LESS WEAR.
- ★ EXCELLENT DOCUMENTATION AND FULL SOFTWARE SUPPORT VIA FUTURE SOFTWARE.
- ★ DISKMON AUTOMATICALLY SUPPORTS COMMERCIAL TYPE PRINTER (CENTRONICS MODELS) VIA PARALLEL PORT.
- ★ DISKDRIVER COMES COMPLETE WITH ATTRACTIVE BEIGE CABINET, BUILT IN POWER SUPPLY, DISK CONTROLLER BOARD, ALL PLUG COMPATIBLE WITH YOUR PET (DISKDRIVER REQUIRES THE EXPANDAPET)
- ★ NOW FOR THE FIRST TIME YOUR PET CAN BE UTILIZED AS A COMPLETE, TOTAL SYSTEM.
- ★ APPLICATION/DEVELOPMENT SOFTWARE AVAILABLE ★ BUSINESS PACKAGES IN MARCH/APRIL.
- ★ FULL PRODUCT (HARDWARE & SOFTWARE) SUPPORT FROM COMPUTHINK AND DEALERS.

HARDWARE/SOFTWARE PRODUCT DIRECTORY

DKH642-1	DISKDRIVER DUAL DRIVE MINIFLOPPY SYSTEM (200K ON LINE STORAGE)	$1295
EXPMEM	EXPANDAPET MEMORY EXPANSION BOARD (SPECIFY 16K, 24K, 32K) 16K MODEL	$ 425
PRT200	CENTRONICS COMMERCIAL PRINTER (MODEL 779-1) WITH FRICTION FEED	$1245
PRT202	CENTRONICS COMMERCIAL PRINTER (MODEL 779-2) WITH TRACTOR FEED	$1345
PRT100	AXIOM HOBBYIST PRINTER (ELECTROSTATIC 5.5" WIDE PAPER) APPROX.	$ 450
NPK101	NEECO PERIPHERAL KEYBOARD (TYPEWRITER TYPE) AVAILABLE JAN.	$ 140
ASG200	PET ASSEMBLER PROGRAMMER'S GUIDE (SHOWS HOW TO TAP INTO BASIC)	$19.95
ASM789D	PET ASSEMBLER 6502 PROGRAMMING SOFTWARE (ON DISK—WITH MANUAL)	$49.95
LNK456D	AUTOLINK PROFESSIONAL LINKING LOADER SOFTWARE (ON DISK—WITH MANUAL)	$49.95
EDT392D	ASSEMBLER-EDITOR, ASSEMBLER SOURCE PROGRAM EDITOR (ON DISK—WITH MANUAL)	$49.95
DUG078	DISKMON USER'S GUIDE (PROVIDED WITH SYSTEM—CAN BE PURCHASED SEPARATELY)	$ 9.95
DKL067	DISKMON (DOS) ASSEMBLER LISTING IN MANUAL FORM	$19.95
FOR112	FORTRAN COMPILER FOR DISKDRIVER SYSTEM (AVAILABLE JAN./FEB.)	$69.95
PLM118	PROFESSIONAL PLM COMPILER FOR DISKDRIVER (AVAILABLE JAN./FEB.)	$49.95
FUTURE	BUSINESS AND PROFESSIONAL PACKAGES STARTING IN JAN./FEB.	

* NOTE PET IS A TRADEMARK OF COMMODORE AND IS SOLD ONLY BY AUTHORIZED DEALERS.

NEECO

EASTERN U.S. & EUROPEAN
NEW ENGLAND ELECTRONICS
248 BRIDGE ST.
SPRINGFIELD, MASS., 01103
(413) 739-9626

COMPUTHINK

N12

WESTERN U.S. & CANADA
COMPUTHINK
3260 ALPINE ROAD
MENLO PARK, CA., 94025

Sherman P. Wantz
424 NW Lakeview Drive
Sebring FL 33870

TRS-80 Level II Reference Manual Index

Level II users who can't "find it" in the manual will find this index invaluable.

Learning to use the TRS-80 Level II BASIC's vocabulary is a lot like trying to master a foreign language. The difficulty involved in learning the many commands, statements, functions, etc., available in Level II BASIC is compounded by the absence of an index in the reference manual that Radio Shack provides.

The table of contents found in the manual is of little use to the novice programmer since it directs the user to chapters that explain the major programming concepts such as strings, arrays, arithmetic functions, etc. But what do you do when the word "REDO" appears on the TRS-80 monitor's screen? Or what response is expected when a double question mark appears?

Fortunately, the answers to these and hundreds of other questions are in the reference manual. The trick is to find them. This index will help you do just that.

Find It Fast

In addition to providing an alphabetical listing of Level II BASIC commands, statements, functions, etc., the index includes a list of symbols (character codes) and indicates where, in the manual, explanations of their uses may be found. The symbols are listed in the order you find them on the TRS-80 computer terminal keyboard reading from left to right and downward, row by row.

To make the index useful, you must number the pages of your Level II reference manual. Mark the first printed sheet of the book (*not* the inside cover) as page one. Page 1 should be the sheet entitled "This Reference Manual and You."

Number the pages of the manual consecutively (including all blank pages) until you reach the page facing the rear cover. This final page—a blank sheet—should be numbered "136." Place numbers (preferably in ink) in the upper or lower corners of each page—or in both places if you wish. Even-numbered pages should appear on your left-side sheets and odd-numbered pages should be on your right.

Here are a few benchmark page numbers to use to be certain that your numbered pages agree with those used in this index:

Now, place appropriate page numbers opposite each line entry found in the table of contents (page 3).

A final step, needed to maximize the use of your manual, involves writing "For Error Codes, see page 115" on page one. Mark pages 115/116 with a paper clip to make them easier to locate. You will be referring to those pages frequently, so make it easy on yourself. ■

Symbols

✓ Reader Service—see page 179

Lewis Tarnopol
164 S. Kingsley Dr.
Los Angeles CA 90004

The Care and Feeding of Cassette Tapes

Part 1 of this article (in December) served as an entrée to cassette tape. Now that you've digested that, it's time to serve the second course: handling techniques and splicing.

In the first part of this series, we determined what kind of cassette to use. Now we will look at proper tape care, splicing and more. Let's start with a not-so-hypothetical situation.

Could This Happen to You?

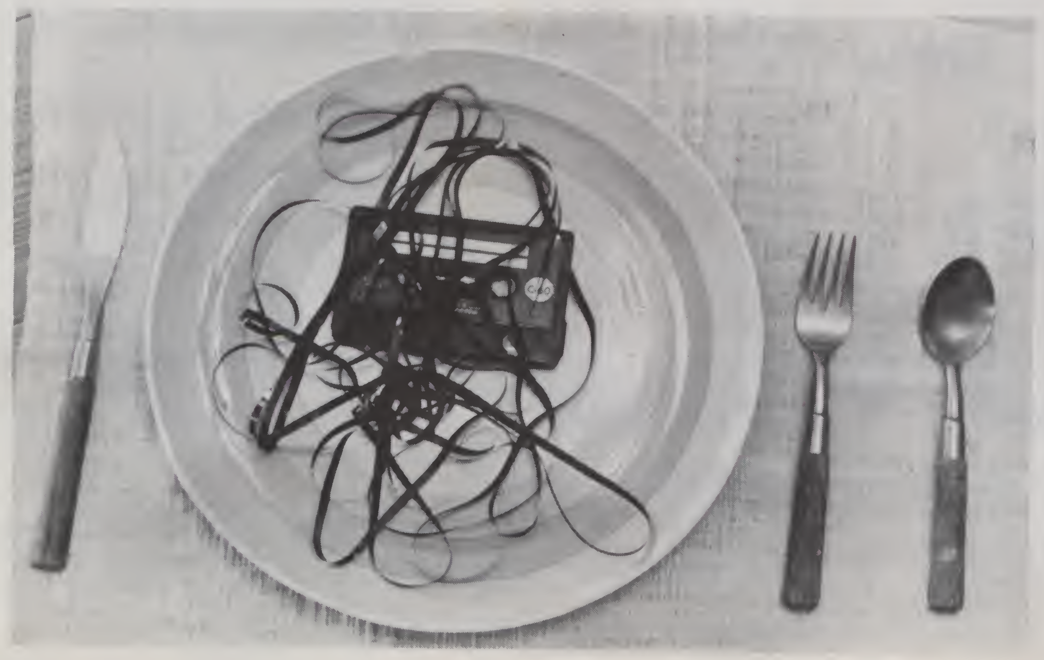
You're a married person with kids that like to touch things, especially cassette recorders. It took you a week to debug a new program, and you've just finished transferring it to cassette. You get up for a rest stop; and while you're gone, one of your children grabs the machine, runs outside to show it to some friends and starts recording—right over your program. You can easily prevent this tragedy from ever occurring by removing a tab insert from the back of the cassette.

Every cassette recorder I've seen has a small metal or plastic sensing device to check if the erase tab has been removed. If it has, you won't be able to record, only play. If you ever do want to record on a detabbed cassette again, simply seal the necessary hole with cellophane tape. There are two tabs that can be removed; each corresponds to only one side of the tape. The correct tab is in the upper *left* corner as you look at the front of the cassette (see Fig. 1).

Some Rules

Modern magnetic tape coatings can retain the intelligence placed on them during the recording process for an infinite amount of time. Age has no weakening effect on what has been recorded; and it is essentially permanent until you erase it, either deliberately or accidentally.

Although the impressed signal will not deteriorate, the physical properties of the cassette tape can be damaged. As a rule, all problems involved with cassette tape are predominantly physical in nature. So it is very important that the tape be handled properly to prevent damage or accidental erasure from occurring. And it is very easy. Here are some rules to follow.



1. Never lay your cassette on your TVT or on any other TV. If you're a ham operator, don't set your headphones on your cassette recorder when there is a tape in it while it's moving. Moving a tape past a magnet (such as the ones in headphones) is a perfect way to erase some of it.

2. To prevent ashes from contaminating your cassettes, don't smoke around them. Hot ashes can ruin a good cassette. The same, of course, goes for food and drink. Liquids also can do your machine in.

3. Try to keep your cassette tapes in a covered area to prevent dust buildup. Dust and dirt can cause the oxide particles to

slowly wear off; and they can prevent good tape-to-head contact.

4. Don't leave your cassettes in the sun or a car glove compartment. Extreme heat buildup might weaken the backing, causing stretching, and sometimes causing the plastic case to melt. If your cassettes are heated by the sun, set them in a cool place. *Do not play them now.* Wait until they cool down or you might stretch them beyond repair.

Tape will allow for low-temperature extremes. (Temperatures as low as -40° F might be encountered in the cargo hold of an aircraft.) Again, allow the tape to achieve equilibrium by waiting for it to reach room temperature and humidity.

If the cassette should become damaged by adverse weather conditions (or you step on it) and you can salvage the tape, there is a fix-it kit sold at hi-fi and electronics-parts stores that contains a new cassette case and splicing tabs. (See the section on splicing for details.)

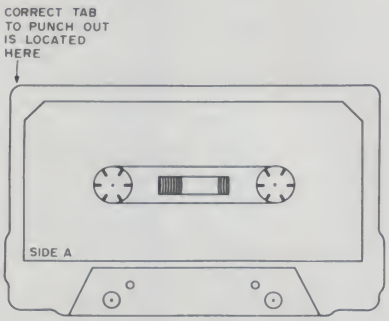


Fig. 1. The correct tab to remove to prevent erasure.

Believe It or Don't

Here are some interesting facts about erasure. Experiments by 3M Company, one of the largest tape manufacturers, have determined that if even a strong magnetic source is at least three or more inches away from the tape, accidental erasure will not occur. Therefore, if you mail any cassette tapes, surround them by at least three inches of packing material. Sturdy packing is a must, anyway, to protect the cassette case from any physical damage. At the airport, your luggage must be exposed to X-rays for weapons' concealment; but never fear—even severe X-rays have *no* effect on magnetic tape. And if you're interested, your cassette tapes will stand up to nuclear radiation up to 200,000 times greater than that which would cause death in 50 percent of the exposed humans.

I also want to make a comment about oxide wear. Unless you purchase poor-quality cassettes, oxide wear is so minimal as to be almost nonexistent. Sure, after 500 plays it's measurable with super-precision measuring instruments; and sure, nothing man makes will last forever. However, considering the modern oxide coatings and the equally well-developed backings they're on, I think the cassette tape of today will far outlast the usefulness you assign to it. If things should ever seem a little less than perfect, you can always make a copy, if you need to.

Splicing

Splicing a cassette tape is considered an art. Whether it's the kind of art you want to practice can be determined by your needs.

If your machine eats the tape and cuts it in half for some reason (e.g., tensions on the machine's take-up and supply motors are unequal), you may want to splice the two pieces back together. You may want to remove a crinkled section from the tape that was eaten but not cut. This is perfectly legitimate as long as the crinkled piece is space between programs.

Suppose you leave your cassette in the glove compartment all day. It doesn't even resemble a cassette now, but the tape inside is still salvageable (you hope it is—it's worth a try to fix it!). Once you get the hang of it, splicing cassettes is easy.

For a professional-type job, you should spend a few bucks and buy a cassette-splicing kit. One company sells a kit called Cassette Saver under the business name of Capitol Tape; it's manufactured by Audio Devices, Inc., Glenbrook CT 06906. The kit contains a new plastic cassette case (you will need this only if the old case is damaged), four splicing tabs, a fancy name for splicing tape, and an edit block, which is used to hold the cassette tape in place while you slice it with a razor blade. It sells for less than four bucks; but for those of you with the spirit, I will describe a do-it-yourself method.

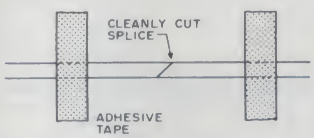


Fig. 2. Splicing the cassette tape (notice the 45° angle).

For any splice to be successful, you *must* use splicing tape—not electrical tape, cellophane tape or anything else. You can buy it anywhere recording tape is sold. Buy whatever thickness you like; you'll have to cut it slightly narrower than the cassette tape width.

Set up your two pieces of tape, backing side facing up, as in Fig. 2 (remove any crinkled tape first). As the drawing indicates, cleanly cut a 45-degree angle in the two pieces of tape you wish to join, butting them so close together you can barely see the crack where they meet. Hold the pieces from moving with adhesive tape. All trimming should be done with a razor blade. Make sure you cut at 45 degrees and not perpendicularly; otherwise, there will be a noise at the splice point when it moves past the playback head.

Next, narrow down a ½ inch piece of splicing tape for the repair and set it gently onto the splice using your fingers or the razor blade. When it's on correctly, there will be no protruding edges (see Fig. 3). Rub it with your fingernail or a cotton swab for a good permanent splice. You are done.

Your tapes should never need a splice job, and you will be able to file the above information with other important things you have learned in the past, such as the population of New Guinea.

Almost to the End

Cassettes will play better in a clean machine. Take a cotton swab, dip it in rubbing alcohol and clean your tape heads, capstan, pinch roller and anything else in the path of the tape. Do it once a month. In the meantime, throw away your old cassettes. They don't contain today's miracle ingredients.

Aside from suggesting that you not allow your pet kangaroo to chew on your cassettes, I think anything else comes under the heading of common sense. I hope you have gained some insight into the care and feeding of cassettes. And may the bird of paradise chirp sweet programs in your ear. ■

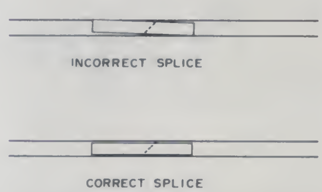


Fig. 3. Incorrect splice (top); correct splice (bottom).

References

Sound Talk Bulletins, Vol. 2, No. 2, 1969; Vol. 3, No. 1, 1970, 3M Company, St. Paul MN.
Robert E. Runstein, *Modern Recording Techniques*, Howard W. Sams and Co., Inc., New York, 1974.
William F. Boyce, *Hi-Fi Stereo Handbook*, Howard W. Sams and Co., Inc., New York, 1967.

SMALL SOFTWARE SYSTEM

TRS-80 SOFTWARE * TRS-80 HARDWARE

SMALL HARDWARE SYSTEM

S51

RSM-1S: A MACHINE LANGUAGE MONITOR FOR 4K TRS-80'S - $23.95

22 commands which interact directly with the Z-80 processor in your TRS-80. Examine your ROM's, test your RAM, enter and execute machine language programs, read and write machine language tapes, and much more! A SYMBOLIC DUMP command disassembles object code and displays it as Zilog standard Z-80 mnemonics! Memory may be displayed in HEX or two ASCII formats, and can be EDITED, MOVED, EXCHANGED, VERIFIED, FILLED, ZEROED, TESTED, or SEARCHED for one or two-byte codes. Memory commands dump continuously or one line at a time! LEVEL-I and II 4K TRS-80's.

RSM-2: AN ADVANCED MONITOR FOR 16K TRS-80'S - $26.95

All the features of our popular RSM-1S, plus read and write SYSTEM tapes, enter Z-80 BREAKPOINTS and PRINT using our TRS232 or the expansion interface! Loads at the top of 16K LEVEL I or II, and includes DISK read and write commands for future expansion!

RSM-2D: 3 MONITORS FOR TRS-80 DISK SYSTEMS - $29.95

RSM-2D contains 3 versions of RSM-2 on a single disk to load into the top of 16K, 32K or 48K TRS-80's with disk systems. With RSM-2D you can read your TRSDOS into memory using our DISK read command, print disassembled listings using our SYMBOLIC dump and printer output commands, then modify and re-write to disk using our DISK write command!

BASIC-1 - LEVEL-1 BASIC FOR LEVEL-II TRS-80'S - $15.95

Loads into the top 4K of 16K TRS-80's and uses any LEVEL-I BASIC program or DATA tape (up to 12K in length) without conversion! Loads from tape or disk (furnished on tape). Go back and forth between LEVEL I and II without reloading. All LEVEL-I abbreviations and functions supported!

AIR RAID: A REAL-TIME TRS-80 SHOOTING GALLERY! - $14.95

A high speed machine language game where large and small airplanes fly across the screen at different altitudes. A ground based missile launcher is pointed and fired from the keyboard. Aircraft explode dramatically when hit, sometimes destroying other nearby planes! Score is tallied for each hit or miss, and the highest score is saved to be challenged by other players. AIR RAID provides hours of fun for you, and is a super demonstration program for entertaining friends! 4K LEVEL I and II.

DCV-1: CONVERT SYSTEM PROGRAMS TO DISK FILES -$9.95

RSM monitors, The Electric Pencil, Air Raid, RSL-1, ESP-1, T-BUG, or nearly any SYSTEM tape shorter than 5K (21K if you have 32K of memory) can now be executed from disk, even if it interferes with TRSDOS! DCV-1 loads system tapes into high memory and adds a block-move routine. TAPEDISK is then used to create a disk file. When accessed from disk, the program loads into high memory, moves itself to its correct address, then jumps there and executes! A super accessory for disk systems!

WORD PROCESSING FOR THE TRS-80 - $99.95

SMALL SYSTEM SOFTWARE has adapted THE ELECTRIC PENCIL to the TRS-80! All standard ELECTRIC PENCIL text editing and printing features, PLUS transparent cursor, two-key rollover, and repeating keyboard. Uppercase only, or add lowercase entry and display with minor modification. PRINT with RS-232 and 20-mil printers using our TRS232 or use the expansion interface. LEVEL-I or II 16K computers. THE ELECTRIC PENCIL is a superior word processor for home or business use!

TRS232 PRINTER INTERFACE - $49.95 (+$2.00 shipping)

A fully assembled self-contained software-driven output port for printing from LEVEL-II, DISK BASIC or machine language programs! Diablo, Teletype, TI Silent or any RS-232 or 20-mil current loop printer may be used. The TRS232 is furnished with cassette software and works either with or without the expansion interface! THE ELECTRIC PENCIL and RSM-2/2D use the TRS232, thus word processing, BASIC, and machine language applications are all supported!

RSL-1: THE GAME OF LIFE IN MACHINE LANGUAGE - $14.95

Enter any pattern with unique repeating keyboard! Save your patterns on tape (4 patterns furnished). Play LIFE, a game of birth, growth and death of a colony of cells. FAST - about 1 second per generation! Hours of fascination, and a super demonstration program!

OTHER TRS-80 PRODUCTS

ESP-1: 29.95 Editor, assembler, and monitor using INTEL 8080 mnemonics.
LST-1: 8.00 A disassembled listing of LEVEL-1 BASIC with some comments.

* CALIFORNIA RESIDENTS ADD 6 PER CENT SALES TAX *

* SMALL SYSTEM SOFTWARE * P.O. BOX 366 * NEWBURY PARK, CALIF. 91320 *

* CALIFORNIA RESIDENTS ADD 6 PER CENT SALES TAX *

* SMALL SYSTEM HARDWARE * P.O. BOX 366 * NEWBURY PARK, CALIF. 91320 *

Julius Caesar tried, Genghis Khan tried, Napoleon tried too . . . now, It's your turn.

We can't guarantee that you'll control the world with a **Mullen Controller Board** kit; but this S-100 peripheral can control disco lights, automatic cat feeders, time lapse photography experiments, environmental heating systems, and a whole lot more. 8 opto-isolator inputs respond to external stimuli . . . your computer decisions based on this information . . . and interfaces to the outside world through 8 on-board computer controlled switches, suitable for driving low power loads or triggering Mullen 500 Watt power modules.

Interested? We thought you would be. So, we've prepared an information packet available at finer computer stores across the nation . . . or, circle the reader information number.

Just what the world needs: products to make your computer useful.

MULLEN Computer Products

BOX 6214, HAYWARD, CA 94545 M32

Reader Service—see page 179

Phil Hughes
PO Box 2847
Olympia WA 98507

Text/Document Preparation Made Easy

For 6800 users who, in effect, type with their elbows, this program, TDOC, is a remedy.

TDOC is a text/document program I developed for producing formatted printed listings from a file created with editor. Why do you want to do that? you ask. This capability is useful for writing letters and articles (particularly if you are a lousy typist). This article was edited using editor and printed using TDOC before submission.

The formatting is handled by two types of commands: One type, which I will call internal control directives, is imbedded in the text and controls such things as new line and new page. The other commands are entered when TDOC is executed and control such entities as line length, lines per page and page numbering. The internal control directives are listed in Table 1, and the execution directives are listed in Table 2.

I have implemented TDOC on an SWTP 6800 system using cassette tape for file storage and a Selectric printer as output device. There is nothing sacred about this hardware configuration since I got the basic idea from the DOC processor on a Univac 1100/42 supercomputer.

Internally TDOC consists of the following major blocks: execution command interpreter, text processor, Selectric print interface and cassette input routines.

The external command interpreter accepts input from the user and sets up data used by the text processor. Entering the start command (carriage return) causes the execution command interpreter to transfer control to the text processor. Control will return when the specified number of pages (from the P command) has been printed.

The text processor reads the cassette tape file using CGET. Using the established line length and page length, it builds output lines and sends them to the printer. Each time a carriage return is encountered the next character is assumed to be an internal control directive and is analyzed. When a page is full, the stop count is decremented. If the stop count is zero, control is returned to the execution command interpreter. Otherwise, the paper is advanced, the heading and page number are printed, if requested, and processing continues.

The Selectric print interface is designed to communicate with a device in Selectric code using an 8-bit parallel port. If you are using a different printer, the Selectric handler can be replaced with any other printer handler. The entry points are as follows:

SINIT—initialize the output port.

SOUT—output the character in A accumulator.

SDATA—output the string

Internal control directives appear as the first character of a text line.

0	Following text is to appear on a new line.
P	Following text is to appear on a new page.
S	Following text is to be single spaced.
D	Following text is to be double spaced.

All other characters are ignored.

Table 1. Internal control directives.

Lnn	Set line length to nn characters.
Pnn	Set lines per page to nn.
Snn	Stop print after nn pages.
R	Reset pointers.
Hheading	Define heading (terminated by carriage return).
Nheading	Define heading with page numbering.
[CR]	Start or continue.

Note: [CR] denotes the ASCII carriage return character.

Table 2. Execution directives.

0 This file is designed to test all
the capabilities of TDOC Version 1.2.
DThe following text is double spaced in order
to use more space and also to verify that
the D command works properly.
S This text is back to single space. This
will verify that the S command will undo
the work of the D command.
0 This stuff starts on a new line.
0 This is also on a new line.
P Finally, this is on a new page.

Example 1. Editor output.

Symbol	Usage
INEEE	Input 1 character from control port.
OUTEEE	Output 1 character to control port.
PDATA	Output data to control port (EOT delimiter).
EPLFC	Number of lines feeds to perform to get from the bottom of current page to the top of next.
CPIA	Address of cassette tape PIA.
SPIA	Address of selectric PIA.
EXIT	Return address to operating system.

Table 3. TDOC configuration dependent parameters.

SETUP WITH L30 S05 P20

This file is designed to test
all the capabilities of TDOC
Version 1.2.

The following text is double

spaced in order to use more

space and also to verify that

the D command works properly.
This text is back to single
space. This will verify that
the S command will undo the
work of the D command.
This stuff starts on a new
line.
This is also on a new line.

SETUP WITH L30 S05 P20

Finally, this is on a new
page.

Example 2. TDOC output.

pointed to by X. Terminated by EOT.

CR—output a carriage return.

SOUT2D—output A accumulator as 2 decimal digits.

The cassette input routines read the editor format cassette and remove the control characters. Two entry points are provided: CINIT—resets buffer pointer for reset command; CGET—gets next input character. If the buffer is empty on a CGET call, the reader is started and then stopped after a block is read. CGET then returns the first character of the block to the user. On all subsequent calls (until the buffer is empty again), CGET returns the next character in the buffer.

Using TDOC is easy. The hardest part for me was getting used to typing both uppercase and lowercase characters on my terminal. This really had nothing to do with TDOC except that I finally had a use for my lowercase characters. Essentially, you just enter the text you want to print, inserting the control information in the first character position of each line. For normal text, such as the body of a letter, this is a blank. If you want to force something to start on a new line, enter a 0 in position 1. For example, to start a new paragraph you would enter 0 in position 1, followed by the number of spaces you wish to indent and the text.

The D and S commands perform the same function as the 0 in position 1, but also set double- or single-spacing mode. For example, if a D occurs in position 1, all text will be double-spaced until an S is encountered in position 1. Save the edited text as you would any source program file and load TDOC. TDOC will prompt with a > when it is ready for commands.

The H command enters a heading of up to 40 characters. This heading will be printed at the top of each page. The N command is equivalent to the H command except page numbers are also printed on the heading line.

The L and P commands determine the page size. L sets the maximum line length. When TDOC is executing it builds the line up to the length set by the L command without breaking a word. In other words, it builds a line of the length specified by L then backs up until it finds a blank. All of the characters following the blank are printed on the next line.

The P command sets the number of lines per page. Once this number is reached, TDOC advances the paper to the next page, prints any requested headings and then continues processing text.

The S command sets a stop after a specific number of pages. If you are printing on something other than continuous forms, setting S to 1 will cause TDOC to stop after every page so a new sheet of paper can be inserted.

The R command resets the buffer pointers and page counter. This is useful after you have done your first page and realize you are printing the wrong tape.

The last command is the start or continue command. I chose the carriage return character for this command because it is possible that the control terminal would also be the printer. Carriage return is a nonprinting character on all systems. If your control terminal is also your printer you could change the prompt character to a bell character (if you have a bell).

Example 1 shows some test data used to verify that TDOC was working properly. Example 2 shows what was produced by TDOC using Example 1 as input. The following execution commands were entered:

L30
S05
P20
HSETUP WITH L30 S05 P20

Table 3 shows the configuration-dependent symbols and their meaning. Note that to be compatible with other SWTP software the cold start address is hex 100 and the restart address is hex 103.

If you get TDOC up and running and are looking for some features to add, consider the following suggestions: justification of the right margin; automatic hyphenation; index and table of contents; center text line; and automatic correction of spelling errors. This should be enough new features for now. ■

Program listing.

```
                         NAM     TDOC
                         OPT     NOG
                         OPT     O
                         OPT     P
                         OPT     NOS
                *
          E1AC  INEEE    EQU     $E1AC
          E1D1  OUTEEE   EQU     $E1D1
          E07E  PDATA    EQU     $E07E
          0004  EPLFC    EQU     4          END PAGE LF COUNT
          8004  CPIA     EQU     $8004
          8018  SPIA     EQU     $8018
                *
                * DIRECT CELLS
0020                     ORG     $20
0020 40         SAVCPL   FCB     64         CHARACTERS PER LINE
0021 3C         SAVLPP   FCB     60         LINES PER PAGE
0022 01         SAVSTP   FCB     1          STOP PAGE COUNT
0023 00         CFLAG    FCB     0          0=START,1=CONT
0024 64         CLINE    FCB     100        CURRENT LINE #
0025 01         STLIN    FCB     1          0=NOT START OF LINE
0026 00         DSFLG    FCB     0          DOUBLE SPACE FLAG
0027 00         CCNUM    FCB     0          CURRENT CHARACTER #
0028 01         PBS      FCB     1          PAGES BEFORE STOP
0029 04         EPFDC    FCB     4          END PAGE LF COUNT
002A 00         PAGE     FCB     0          CURRENT PAGE #
002B 00         SBFLAG   FCB     0          1=SKIP BLANKS
```

```
002C 00        EOFFLG FCB    0         EOF FLAG
002D 0002      TSAVX  RMB    2
002F 0002      CSAVX  RMB    2         CIO SAVE X
0031 0002      NLPTR  RMB    2         NEW LINE PTR
0033 0002      SAVX   RMB    2         X SAVE FOR SIO
0035 0002      CDPTR  RMB    2         CASSETTE DATA POINTER
0037 0002      LPTR   RMB    2         LBUF PTR
0039 0002      LSAVX  RMB    2         LPUT SAVE X
               *
0100                  ORG    $100
0100 7E 0243          JMP    START
0103 7E 0249          JMP    CLOOP
0106 7E 2E00   EXIT   JMP    $2E00     RETURN TO O.S.
0120                  ORG    $120
               *
               * DATA
0120 20        HMSG   FCC    40,
0148 04               FCB    $04       EOT
0149 50        PMSG   FCC    /PAGE /
014E 04               FCB    $04       EOT
               * CASSETTE BUFFER
014F 0080      CASBUF RMB    128
01CF 0064      LBUF   RMB    100       LINE BUFFER
0233 44        STMSG  FCC    /DOC - V1.2 /
023E 04               FCB    $04
023F 0D0A      PMTMSG FDB    $0D0A
0241 3E               FCC    />/
0242 04               FCB    $04
               *
     0243      START  EQU    *
0243 CE 0233          LDX    #STMSG
0246 BD E07E          JSR    PDATA
               * INTERPRET CONTROL COMMANDS FROM TERMINAL
     0249      CLOOP  EQU    *         CONTROL LOOP
0249 CE 023F          LDX    #PMTMSG
024C BD E07E          JSR    PDATA
024F BD E1AC          JSR    INEEE     GET COMMAND
0252 81 4C            CMP A  #'L       LINE LENGTH?
0254 27 56            BEQ    CPL
0256 81 50            CMP A  #'P       PAGE LENGTH
0258 27 4A            BEQ    LPP
025A 81 48            CMP A  #'H       HEADING?
025C 27 5D            BEQ    HEAD
025E 81 53            CMP A  #'S       SET STOPS
0260 27 75            BEQ    STOPS
0262 81 52            CMP A  #'R       RESET
0264 27 0E            BEQ    RESET
0266 81 4E            CMP A  #'N       PAGE NUMBER
0268 27 4D            BEQ    NHEAD
026A 81 0D            CMP A  #$0D      RUN?
026C 27 0B            BEQ    RUN
026E 81 58            CMP A  #'X       EXIT
0270 26 D7            BNE    CLOOP     TRY AGAIN
0272 20 40            BRA    EOFR
               * RESET
     0274      RESET  EQU    *
0274 7F 0023          CLR    CFLAG     RESET CONT FLAG
0277 20 D0            BRA    CLOOP     GET NEXT COMMAND
               * START/CONTINUE
     0279      RUN    EQU    *
0279 96 22            LDA A  SAVSTP
027B 97 28            STA A  PBS       PAGES BEFORE STOP
027D 96 23            LDA A  CFLAG     START/CONT?
027F 26 20            BNE    CONT
               * START
0281 86 64            LDA A  #100
0283 97 24            STA A  CLINE
0285 97 25            STA A  STLIN
0287 CE 01CF          LDX    #LBUF
028A DF 37            STX    LPTR
028C 5F               CLR B            CLEAR
028D D7 27            STA B  CCNUM     CURR CHAR
028F D7 2B            STA B  SBFLAG    & S.B. FLAG
0291 D7 2C            STA B  EOFFLG    CLEAR EOF FLAG
0293 D7 26            STA B  DSFLG     CLEAR DOUBLE SPACE
               * INITIALIZATION
0295 BD 040D          JSR    SINIT     SELECTRIC
0298 BD 050A          JSR    CINIT     CASSETTE
029B 7C 0023          INC    CFLAG     SET CONTINUE FLAG
029E 7E 03BA          JMP    HSTRT     FORCE HDG
     02A1      CONT   EQU    *
               * REAL PRINT LOOP
02A1 7E 03BA          JMP    HSTRT     CONTINUE
               *
               * SET LINES PER PAGE
     02A4      LPP    EQU    *
02A4 BD 02DF          JSR    GET2D     2 DIGIT DEC
02A7 97 21            STA A  SAVLPP
02A9 7E 0249          JMP    CLOOP
               *
               * SET CHARS PER LINE
     02AC      CPL    EQU    *
02AC BD 02DF          JSR    GET2D
02AF 97 20            STA A  SAVCPL
02B1 7E 0249          JMP    CLOOP
               *
     02B4      EOFR   EQU    *
02B4 7E 0106          JMP    EXIT
               *
               * SAVE HEADING
     02B7      NHEAD  EQU    *
02B7 86 01            LDA A  #1
02B9 97 2A            STA A  PAGE
     02BB      HEAD   EQU    *
02BB CE 0120          LDX    #HMSG
02BE C6 28            LDA B  #40
02C0 BD E1AC   H100   JSR    INEEE
02C3 81 0D            CMP A  #$0D      CR?
02C5 27 08            BEQ    HEND      YES
02C7 A7 00            STA A  0,X       STORE CHAR
02C9 08               INX
02CA 5A               DEC B
02CB 2E F3            BGT    H100      40 CHARS?
```


✓ Reader Service—see page 179

```
02CD 27 0D     HBLUP  BEQ    XCL
02CF 86 20     HEND   LDA A  #'
02D1 A7 00            STA A  0,X
02D3 08               INX
02D4 5A               DEC B
02D5 20 F6            BRA    HBLUP
                      *
                      * SET PAGE STOPS
     02D7      STOPS  EQU    *
02D7 BD 02DF          JSR    GET2D
02DA 97 22            STA A  SAVSTP
     02DC      XCL    EQU    *
02DC 7E 0249          JMP    CLOOP
                      *
                      * GET 2 DECIMAL DIGITS
     02DF      GET2D  EQU    *
02DF BD E1AC          JSR    INEEE    TENS
02E2 80 30            SUB A  #$30     TO DECIMAL
02E4 48               ASL A           *10
02E5 16               TAB
02E6 48               ASL A
02E7 48               ASL A
02E8 1B               ABA
02E9 16               TAB
02EA BD E1AC          JSR    INEEE    UNITS
02ED 80 30            SUB A  #$30     TO DEC
02EF 1B               ABA             + TENS
02F0 39               RTS
                      *
     02F1      PLINC  EQU    *
02F1 86 64            LDA A  #100
02F3 97 24            STA A  CLINE
02F5 20 45            BRA    NXL
                      *
                      * GET NEXT TEXT CHARACTER
     02F7      GETNXT EQU    *
02F7 7D 002C          TST    EOFFLG   DID WE GET EOF
02FA 26 B8            BNE    EOFR     YES
02FC BD 0513          JSR    CGET     GET NEXT CHARACTER
02FF 81 1A     PL000  CMP A  #$1A     EOF?
0301 26 05            BNE    PL005    NO
0303 7C 002C          INC    EOFFLG   YES-SET EOF FLAG
0306 20 34            BRA    NXL      PRINT LAST LINE
     0308      PL005  EQU    *
0308 81 0D            CMP A  #$0D     CR?
030A 26 06            BNE    PL010    NO
030C 97 25            STA A  STLIN    SET STLIN
030E 86 20            LDA A  #'
0310 20 34            BRA    PL100
0312 81 20     PL010  CMP A  #$20     OTHER CONTROL?
0314 2D E1            BLT    GETNXT   IGNORE
0316 7D 0025          TST    STLIN    START OF LINE?
0319 27 2B            BEQ    PL100    NO
031B 7F 0025          CLR    STLIN    CLEAR SL FLAG
031E 81 20            CMP A  #'       NORMAL TEXT?
0320 27 D5            BEQ    GETNXT   YES
0322 81 50            CMP A  #'P      PAGINATE?
0324 27 CB            BEQ    PLINC
0326 81 44            CMP A  #'D      DOUBLE SPACE?
0328 26 05            BNE    PL020    NO
032A 7C 0026          INC    DSFLG    YES-SET FLAG
032D 20 0D            BRA    NXL      AND DO NEXT LINE
032F 81 53     PL020  CMP A  #'S      SINGLE SPACE?
0331 26 05            BNE    PL030    NO
0333 7F 0026          CLR    DSFLG    YES-CLEAR FLAG
0336 20 04            BRA    NXL      AND DO NEXT LINE
     0338      PL030  EQU    *
0338 81 30            CMP A  #'0      NEW LINE?
033A 26 BB            BNE    GETNXT   INVALID
     033C      NXL    EQU    *
033C 86 04            LDA A  #4       EOT
033E BD 03EF          JSR    LPUT
0341 4F               CLR A           SBFLAG
0342 5F               CLR B           MOVE COUNT
0343 7E 0371          JMP    LIF200
     0346      PL100  EQU    *
0346 81 20            CMP A  #'       BLANK?
0348 26 05            BNE    PL120    NO
034A 7D 002B          TST    SBFLAG   SKIP BLANKS?
034D 26 A8            BNE    GETNXT   YES
     034F      PL120  EQU    *
034F 7F 002B          CLR    SBFLAG   CLEAR SKIP FLAG
0352 BD 03EF          JSR    LPUT     SAVE CHAR IN LBUF
0355 D6 20            LDA B  SAVCPL   CHAR PER LINE
0357 D0 27            SUB B  CCNUM    LINE FULL?
0359 2D 02            BLT    LINFUL   YES
035B 20 9A            BRA    GETNXT
                      * LINE FULL
     035D      LINFUL EQU    *
035D 5F               CLR B           BACKUP COUNT
035E DE 37            LDX    LPTR     NEXT LOC IN LBUF
0360 09        LIF020 DEX             BACKUP
0361 A6 00            LDA A  0,X      GET CHAR
0363 81 20            CMP A  #'       BLANK?
0365 27 03            BEQ    LIF100   YES
0367 5C               INC B           BUMP BACKUP COUNT
0368 20 F6            BRA    LIF020   TRY AGAIN
     036A      LIF100 EQU    *
036A 86 04            LDA A  #$04     EOT
036C A7 00            STA A  0,X      SAVE
036E 08               INX             NEXT LINE ADDRESS
036F DF 31            STX    NLPTR    SAVE
     0371      LIF200 EQU    *
0371 97 2B            STA A  SBFLAG   SET SBFLAG
0373 CE 01CF          LDX    #LBUF    BUFFER START
0376 DF 37            STX    LPTR     RESET
0378 BD 0401          JSR    SDATA    PRINT LINE
037B BD 0408          JSR    CR
037E 7F 0027          CLR    CCNUM    RESET CHAR COUNT
0381 5D               TST B
0382 27 0E            BEQ    LIF300   NO DATA TO MOVE
0384 7F 002B          CLR    SBFLAG   NOT START OF LINE
                      * MOVE TO START OF LBUF
```

```
0387 DE 31          LDX    NLPTR
0389 A6 00   LIF250 LDA A  0,X
038B 08             INX
038C BD 03EF        JSR    LPUT
038F 5A             DEC B
0390 26 F7          BNE    LIF250
     0392    LIF300 EQU    *
0392 7C 0024        INC    CLINE
0395 7D 0026        TST    DSFLG    DOUBLE SPACE?
0398 27 06          BEQ    LIF400   NO
039A 7C 0024        INC    CLINE    YES
039D BD 0408        JSR    CR
             * PAGE FULL?
     03A0    LIF400 EQU    *
03A0 D6 21          LDA B  SAVLPP
03A2 D0 24          SUB B  CLINE    PAGE FULL?
03A4 2B 03          BMI    NXTPGE   YES
03A6 7E 02F7        JMP    GETNXT   NO
     03A9    NXTPGE EQU    *
03A9 96 28          LDA A  PBS
03AB 4A             DEC A
03AC 26 03          BNE    NOSTOP
03AE 7E 0249        JMP    CLOOP
03B1 97 28   NOSTOP STA A  PBS
             * LF TO NEXT PAGE
03B3 C6 04          LDA B  #EPLFC
03B5 8D 51   LFLUP  BSR    CR
03B7 5A             DEC B
03B8 2A FB          BPL    LFLUP
     03BA    HSTRT  EQU    *
03BA DF 2D          STX    TSAVX
03BC CE 0120        LDX    #HMSG
03BF BD 0401        JSR    SDATA
03C2 96 2A          LDA A  PAGE     NEXT PAGE NUMBER
03C4 27 1C          BEQ    NOPGN
03C6 D6 20          LDA B  SAVCPL   LINE LENGTH
03C8 C0 28          SUB B  #40      -HDG LENGTH
03CA 2B 16          BMI    NOPGN    NO ROOM FOR PAGE #
03CC 86 20   SPLUP  LDA A  #'
03CE BD 0420        JSR    SOUT
03D1 5A             DEC B
03D2 2A F8          BPL    SPLUP
03D4 CE 0149        LDX    #PMSG    PAGE
03D7 BD 0401        JSR    SDATA
03DA 96 2A          LDA A  PAGE
03DC BD 0438        JSR    SOUT2D   OUT 2 DIGITS
03DF 7C 002A        INC    PAGE
03E2 8D 24   NOPGN  BSR    CR
03E4 8D 22          BSR    CR
03E6 DE 2D          LDX    TSAVX
03E8 86 01          LDA A  #1
03EA 97 24          STA A  CLINE
03EC 7E 02F7        JMP    GETNXT
             *
             * PUT NEXT CHAR IN LINE BUFFER
     03EF    LPUT   EQU    *
03EF DF 39          STX    LSAVX
03F1 DE 37          LDX    LPTR
03F3 A7 00          STA A  0,X
03F5 08             INX
03F6 DF 37          STX    LPTR
03F8 DE 39          LDX    LSAVX
03FA 7C 0027        INC    CCNUM    CHAR COUNT
03FD 39             RTS
             *
             *
             *
             * SELECTRIC PRINT
03FE 8D 20   SDATA1 BSR    SPRINT
0400 08             INX
             *
     0401    SDATA  EQU    *
0401 A6 00          LDA A  0,X
0403 81 04          CMP A  #4
0405 26 F7          BNE    SDATA1
0407 39             RTS
             *
     0408    CR     EQU    *        PRINT CR
0408 86 0D          LDA A  #$0D
040A 8D 14          BSR    SPRINT
040C 39             RTS
             *
040D 86 FF   SINIT  LDA A  #$FF
040F B7 8018        STA A  SPIA
0412 86 2E          LDA A  #$2E
0414 B7 8019        STA A  SPIA+1
0417 86 2C          LDA A  #$2C
0419 B7 8018        STA A  SPIA
041C B6 8018        LDA A  SPIA
041F 39             RTS
             *
             *
     0420    SOUT   EQU    *
     0420    SPRINT EQU    *
0420 DF 33          STX    SAVX
0422 7D 8019 SWAIT  TST    SPIA+1
0425 2A FB          BPL    SWAIT
0427 CE 0456        LDX    #ASCCOR
042A B7 042E        STA A  IINX
     042E    IINX   EQU    *+1
042D A6 00          LDA A  0,X
042F B7 8018        STA A  SPIA
0432 B6 8018        LDA A  SPIA
0435 DE 33          LDX    SAVX
0437 39             RTS
             * OUTPUT A AS 2 DECIMAL DIGITS
     0438    SOUT2D EQU    *
0438 37             PSH B
0439 36             PSH A
043A C6 2F          LDA B  #$2F
     043C    SL100  EQU    *
043C 5C             INC B
043D 80 0A          SUB A  #10
043F 2A FB          BPL    SL100
0441 17             TBA
0442 BD 0420        JSR    SOUT
0445 C0 30          SUB B  #$30
0447 58             ASL B
0448 17             TBA
0449 58             ASL B
044A 58             ASL B
044B 1B             ABA
044C 17             TBA
044D 32             PUL A
044E 10             SBA
044F 8B 30          ADD A  #$30
0451 BD 0420        JSR    SOUT
0454 33             PUL B
0455 39             RTS
             *
             *
             *
     0072    Z      EQU    $72      CENT SIGN
     0456    ASCCOR EQU    *        ASCII TO SELECTRIC TABLE
0456 72             FCB    Z,Z,Z,Z,Z,Z,Z,Z,Z
045F 0C             FCB    $0C      TAB
0460 1C             FCB    $1C      LF
0461 72             FCB    Z,Z
0463 2C             FCB    $2C      CR
0464 72             FCB    Z,Z,Z,Z,Z,Z,Z,Z,Z,Z
046E 72             FCB    Z,Z,Z,Z,Z,Z,Z,Z
0476 3C             FCB    $3C      SP
0477 7F             FCB    $7F,$6A,$77,$79,$7A,$FB,$2A  ! " # $
047E 70             FCB    $70,$78,$73,$46  ( ) * +
0482 03             FCB    $03,$00,$26,$09  , - . /
0486 38             FCB    $38,$3F,$36,$37  0 1 2 3
048A 39             FCB    $39,$3A,$32,$3B  4 5 6 7
048E 33             FCB    $33,$30,$4B,$0B  8 9 : ;
0492 43             FCB    $43,$06,$66,$49  , = . ?
0496 76             FCB    $76,$63,$50,$53  @ A B C
049A 5B             FCB    $5B,$5A,$47,$4F  D E F G
049E 58             FCB    $58,$62,$4E,$52  H I J K
04A2 59             FCB    $59,$6F,$56,$69  LM N O
04A6 4A             FCB    $4A,$42,$6B,$68  P Q R S
04AA 5E             FCB    $5E,$57,$67,$60  T U V W
04AE 5F             FCB    $5F,$48,$71,$6E  X Y Z [
04B2 72             FCB    Z,$2E,$40,Z  ¢ ] ▵ ¢
04B6 72             FCB    Z,$23,$10,$13  ¢ a b c
04BA 1B             FCB    $1B,$1A,$07,$0F  d e f g
04BE 18             FCB    $18,$22,$0E,$12  h i j k
04C2 19             FCB    $19,$2F,$16,$29  l m n o
04C6 0A             FCB    $0A,$02,$2B,$28  p q r s
04CA 1E             FCB    $1E,$17,$27,$20  t u v w
04CE 1F             FCB    $1F,$08,$31,Z  x y z
04D2 72             FCB    Z,Z,Z,Z
             *
             * CASSETTE INPUT ROUTINES
             *
     04D6    CREAD  EQU    *        READ 128 BYTE BLOCK
04D6 86 11          LDA A  #$11
04D8 BD E1D1        JSR    OUTEEE
04DB 86 3C          LDA A  #$3C
04DD B7 8007        STA A  CPIA+3
             * LOOK FOR S9
04E0 BD E1AC RDADR  JSR    INEEE
04E3 81 53          CMP A  #'S
04E5 26 F9          BNE    RDADR
04E7 BD E1AC        JSR    INEEE
04EA 81 39          CMP A  #'9
04EC 26 F2          BNE    RDADR
             * GET DATA
04EE 37             PSH B
04EF 5F             CLR B
04F0 BD E1AC RDCHR  JSR    INEEE
04F3 A7 00          STA A  0,X
04F5 08             INX
04F6 5C             INC B
04F7 C1 80          CMP B  #$80
04F9 26 F5          BNE    RDCHR    GET NEXT CHAR
04FB BD E1AC        JSR    INEEE
             * STOP READER
04FE 86 13          LDA A  #$13
0500 BD E1D1        JSR    OUTEEE
0503 86 34          LDA A  #$34
0505 B7 8007        STA A  CPIA+3
0508 33             PUL B
0509 39             RTS
             * RESET CASSETTE DATA POINTER
     050A    CINIT  EQU    *
050A CE 014F        LDX    #CASBUF
050D DF 35          STX    CDPTR
050F BD 04D6        JSR    CREAD
0512 39             RTS
             *
             * GET CHARACTER FROM CASSETTE
     0513    CGET   EQU    *
0513 DF 2F          STX    CSAVX
0515 DE 35          LDX    CDPTR
0517 A6 00          LDA A  0,X
0519 08             INX
051A 8C 01CF        CPX    #CASBUF+128
051D 27 05          BEQ    CBMPTY
051F DF 35          STX    CDPTR
0521 DE 2F          LDX    CSAVX
0523 39             RTS
0524 CE 014F CBMPTY LDX    #CASBUF
0527 DF 35          STX    CDPTR
0529 36             PSH A
052A 8D AA          BSR    CREAD    FILL BUFFER
052C 32             PUL A
052D 39             RTS
             *
A048                ORG    $A048
A048 0100           FDB    $0100
                    END
TOTAL ERRORS 00000
```

INTRODUCING

DUAL DRIVE MINIFLOPPY FOR PET!

DEALER INQUIRIES INVITED

DKH641 IS A PRODUCT OF COMPUTHINK

- •DUAL MINI FLOPPY DRIVE WITH 100K PER DISK SIDE FOR TOTAL 200K ON LINE.
- •DESIGNED FOR COMMERCIAL SPEED REQUIREMENTS •FAST LOADING SPEED.
- •DISKMON™ (DOS) AUTOMATICALLY REORGANIZES FREE DISK SPACE AFTER SAVE OR ERASE.
- •DISKMON IS RESIDENT IN ROM VIA DISK CONTROLLER BOARD PLUGGED INTO EXPANDAPET.•
- •DISKMON ADDS 16 COMMANDS TO BASIC INCLUDING DISK DATA FILES.
- •DISKMON COMMANDS SUPPORT COMMERCIAL PRINTER OFF PARALLEL PORT SUCH AS CENTRONICS 779.
- •FULL DISK SOFTWARE SUPPORT • FORTRAN & PLM COMPILERS THIS JANUARY.
- •90 DAY MANUFACTURER'S WARRANTY ON HARDWARE•READY TO USE ON DELIVERY. WITH FULL INSTRUCTIONS AND UTILITY DISKETTE.
- •CALL OR WRITE FOR ADDITIONAL INFORMATION • INITIAL QUANTITIES LIMITED.
- •THIS SYSTEM IS FULLY SUPPORTED BY SOFTWARE FROM THE MANUFACTURER.
- •ORDERS ARE NOW BEING ACCEPTED FOR DECEMBER & JANUARY DELIVERY.

DKH642	—DUAL DRIVE SYSTEM, COMPLETE WITH DISKMON	$1295.00
DKL067	—DISKMON ASSEMBLER LISTING/DOS	$19.95
ASM789T	—PET ASSEMBLER ON CASSETTE	$19.95
ASM789D	—PET ASSEMBLER ON DISKETTE (5.5 inch)	$49.95
LNK456	—AUTOLINK LINKING LOADER ON DISKETTE	$49.95
FOR300	—FORTRAN COMPILER ON DISKETTE (JAN'79)	$69.95
PLM400	—PLM COMPILER ON DISKETTE (JAN'79)	$49.95

★ ★ ★ BUSINESS PACKAGES STARTING IN 1st QUARTER 1979 ★ ★ ★

•THIS SYSTEM REQUIRES EXPANDAPET MEMORY (MINIMUM 16K—SEE BELOW)

PET COMPUTER

IF YOU HAVEN'T BOUGHT A PET YET, CALL FOR OUR PET INFO PACKAGE

WHY NOT BUY FROM THE BEST?

8K PET	$ 795
24K PET (8 + 16K)	$1220
32K PET (8 + 24K)	$1320

ALL PRICES INCLUDE 48 HR. PRE-SHIPMENT TESTING & 3 FREE CASSETTE PROGRAMS

PRICES SHOWN ABOVE INCLUDE EXPANDAPET. PME MEMORIES WILL HAVE HIGHER PRICING.

- ★ NEECO IS A CUSTOMER ORIENTED, FULL SERVICE COMPANY.
- ★ PETS RECEIVE 48 HR. 'BURNIN' BY NEECO BEFORE SHIPMENT.
- ★ FULL CUSTOMER SERVICE AND FULL PRODUCT SUPPORT.
- ★ 48 HR MAXIMUM 'TURNAROUND' ON PET WARRANTY SERVICE ON PETS PURCHASED FROM NEECO.
- ★ FULL PRE-PURCHASE INFO AVAILABLE FROM OUR PET INFO PACKAGE — WE ANSWER CUSTOMER QUESTIONS!
- ★ AUTOMATIC SOFTWARE/HARDWARE UPDATES VIA OUR PET OWNERS MAILING LIST — CALL/WRITE TO BE LISTED!
- ★ COMMERCIAL QUANTITIES AVAILABLE.
- ★ WE ALSO MARKET REPLACEMENT RAMS & ROMS, ETC.
- ★ OFF THE SHELF DELIVERIES (NO DEPOSIT REQUIRED).

SCHOOL INQUIRIES INVITED!

INTERNAL MEMORY EXPANSION FOR PET!

EXPANDAPET™

INTERNAL MEMORY EXPANSION UNIT

DEALER INQUIRIES INVITED

- •MOUNTS EASILY INSIDE YOUR PET
- •EASY TO INSTALL (15 MINUTES)
- •NO DEGRADATION OF PET SYSTEM
- •USES LOW POWER DYNAMIC RAMS
- •90 DAY PART&LABOR, 1 YR-RAMS.
- •30 DAY MONEY BACK GUARANTEE.
- •MOUNTING SLOTS FOR 4 BOARDS.
- •CALL/WRITE FOR ADDITIONAL INFO
- •DEALER INQUIRES INVITED.

EXPANDAPET PRICES

16K (+ 8K PET = 24K)	$425
24K (+ 8K PET = 32K)	$525
32K (+ 8K PET = 40K)	$615

32K UNIT ALLOWS 8K OF ASSEMBLY LANGUAGE SUBROUTINES ACCESSED VIA THE USR COMMAND.

OPTIONAL PLUG-IN BOARDS

SERIAL I/O BOARD	$75
S-100 I/O BOARD	$75
4K EPROM BOARD	$50

EXPANDAPET IS A PRODUCT OF COMPUTHINK

FULLSIZED TYPEWRITER KEYBOARD FOR PET!

- •COMMERCIAL QUALITY KEYBOARD WITH METAL ENCLOSURE.
- •BASIC TYPEWRITER DESIGN FOR TOUCHTYPISTS.
- •SINGLE KEY FUNCTIONS FOR ALL CURSOR CONTROLS. SHIFT/RUN, INSERT, CLEAR SCREEN/HOME CURSOR, MORE.
- •FUNCTIONS SIMULTANEOUSLY WITH PET'S KEYBOARD.
- •PLUGS DIRECTLY INTO PET'S LOGIC BOARD.
- •DOES NOT USE USER OR IEEE-488 PORTS.
- •NPK-101 IS FULLY TESTED & READY TO USE.
- •ATTACHES DIRECTLY TO FRONT OF PET'S FRAME.
- •CAN BE USED AS A REMOTE TERMINAL (SPECIAL ORDER).
- •30 DAYS TRIAL PERIOD • 90 DAY WARRANTY.
- •CALL OR WRITE FOR FULL SPECS-INITIAL QTY LIMITED.

ORDERS ARE NOW BEING ACCEPTED FOR DEC/JAN CUSTOMER DELIVERIES. **$139.95**

NPK-101 IS A PRODUCT OF NEW ENGLAND ELECTRONICS.

WE CANNOT LIST ALL OF OUR SOFTWARE AND HARDWARE PRODUCTS CALL OR WRITE FOR OUR *FREE* SOFTWARE/HARDWARE DIRECTORY

DOMESTIC & OVERSEAS DEALER INQUIRIES INVITED ON • MEMORY • KEYBOARD • FLOPPY

NEECO

NEW ENGLAND ELECTRONICS CO., INC.
248 BRIDGE ST., SPRINGFIELD, MASS. 01103
MON-FRI 9:30-5:30, SAT 10-2, E.S.T.

N12

(413) 739-9626

master charge VISA ACCEPTED, ADD 3% SERVICE CHARGE.

Simpler Interest

When you're calculating mortgage payments, a simpler way can be a better way.

Rod Hallen
Road Runner Ranch
PO Box 73
Tombstone AZ 85638

It used to be that I'd get a receipt back each month after I made my mortgage payment. Listed on it would be the amount of my old balance, payment made, principal, interest and new balance. Then the title company put in a computer, and now I get an annual report each January and that's all.

Here is an opportunity for my computer to give me what the company's won't. "Simpler Interest" will ask for the old balance, annual interest rate, amount of the payment and how many months you want printed out. It will then compute and display (or print hard copy) the month number, payment, principal, interest, new balance and the total interest paid. If you enter the balance of January 1st and ask for 12 months, you'll have a complete picture for the next year.

If the balance reaches zero within the requested time period, the total months to pay off and the amount of the last month's payment will be

```
100 PRINT "SIMPLER INTEREST"
110 PRINT "BY ROD HALLEN  TOMBSTONE, ARIZONA  18 DEC 77"
120 PRINT
130 INPUT "BALANCE ? $",B
140 PRINT
150 INPUT "INTEREST RATE ? %",I
160 PRINT
170 INPUT "PAYMENT ? $", P
180 PRINT
190 INPUT "MONTHS ?", A
200 PRINT
210 PRINT "MONTH","PAYMENT","PRINCIPAL","INTEREST","NEW BALANCE"
220 PRINT %2%
230 Z=Ø
240 FOR M=1TOA
250 X=B*(I/100)/12
260 Y=P-X
270 Z=Z+X
280 IF B<=P THEN GOTO 330
290 B=B-Y
300 PRINT M,P,Y,X,B
310 NEXT M
320 GOTO 350
330 PRINT "LOAN PAID OFF IN";M;"MONTHS WITH THE FINAL"
340 PRINT "MONTHS PAYMENT BEING $"; B
350 PRINT "TOTAL INTEREST PAID $"; Z
360 END
```

Program listing for "Simpler Interest." It is written in Processor Technology's BASIC 5 but might need some modifications to fit your BASIC. Line 220 causes BASIC 5 to print numbers to two decimal places; leave it out if it doesn't work. If you're not allowed to input "BALANCE, B," then make line 130 PRINT "BALANCE" and line 135 INPUT B. Ditto for the rest of the INPUTs. When entering figures such as $45,000, do not type the $ or the comma. Interest is entered without the %. 7.5% would be typed 7.5. Some BASICs will require LET in statements on lines 230, 250, 260, 270 and 290. All of this can be shortened considerably if you can place more than one statement on a line. This is OK with BASIC 5, but I find it easier to rewrite and change my program if I stick to one statement per line.

```
SIMPLER INTEREST
BY ROD HALLEN  TOMBSTONE, ARIZONA  18 DEC 77

BALANCE ? $987.5Ø

INTEREST RATE ? %8.5

PAYMENT ? $45.67

MONTHS ?12

MONTH       PAYMENT     PRINCIPAL    INTEREST    NEW BALANCE

 1          45.67       38.68        6.99        948.82
 2          45.67       38.95        6.72        9Ø9.88
 3          45.67       39.23        6.44        87Ø.65
 4          45.67       39.5         6.17        831.15
 5          45.67       39.78        5.89        791.36
 6          45.67       4Ø.Ø6        5.61        751.3
 7          45.67       4Ø.35        5.32        71Ø.95
 8          45.67       4Ø.63        5.Ø4        67Ø.32
 9          45.67       4Ø.92        4.75        629.4
 1Ø         45.67       41.21        4.46        588.18
 11         45.67       41.5         4.17        546.68
 12         45.67       41.8         3.87        5Ø4.88
TOTAL INTEREST PAID $ 65.42
```

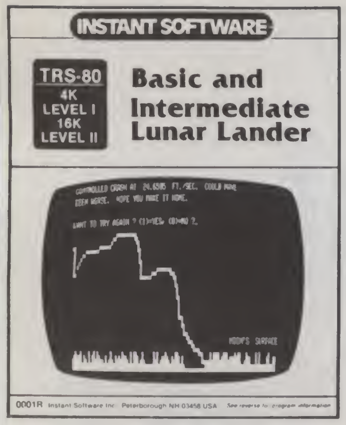
Fig. 1. This is a 12-month run on a typical loan. The balance on January 1st was $987.50, the interest rate is 8.5% and the monthly payment is $45.67. At the end of the year we still owe $504.88 and during the year we paid $65.42 in interest.

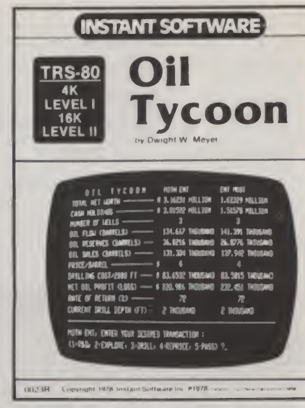
shown. By entering a very large number for the month's input, the program will run until payoff. This way you always know where you stand.

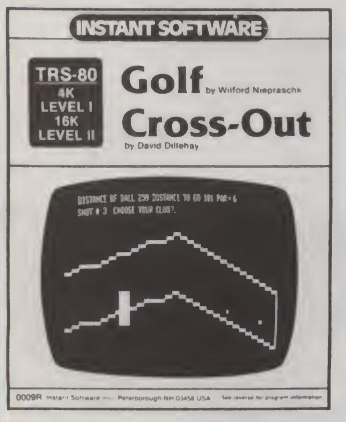
If you are allowed to increase your payments without penalty, it is interesting to see how much interest you can save and how much quicker your loan is paid off by adding $5 or $10 a month to your payment.■

```
SIMPLER INTEREST
BY ROD HALLEN  TOMBSTONE, ARIZONA  18 DEC 77

BALANCE ? $987.50

INTEREST RATE ? %8.5

PAYMENT ? $95.23

MONTHS ?12

MONTH       PAYMENT      PRINCIPAL    INTEREST     NEW BALANCE

 1           95.23        88.24        6.99         899.26
 2           95.23        88.86        6.37         810.4
 3           95.23        89.49        5.74         720.91
 4           95.23        90.12        5.11         630.79
 5           95.23        90.76        4.47         540.03
 6           95.23        91.4         3.83         448.62
 7           95.23        92.05        3.18         356.57
 8           95.23        92.7         2.53         263.87
 9           95.23        93.36        1.87         170.51
 10          95.23        94.02        1.21         76.49
LOAN PAID OFF IN 11MONTHS WITH THE FINAL
MONTHS PAYMENT BEING $ 76.49
TOTAL INTEREST PAID $ 41.83
```

Fig. 2. This run is similar to Fig. 1 except that this loan would be paid off within the period of months that was entered.







*IMP™ for TRS-80™

*Instant Micro Programs

Level I on one side—Level II on the other

• AIR FLIGHT SIMULATION by Dwight W. Meyer (for the TRS-80™) Your mission is to take off and land your aircraft without crashing. You're flying blind on instruments only. The flight starts with a full tank of fuel giving you approximately 50 miles of range. The computer constantly updates your air speed, compass heading and altitude. As you grow more sure of yourself in the pilot seat, you can fly a course against a map or do acrobatic maneuvers. $7.95. Order no. 0017R.

• BASIC LUNAR LANDER/INTERMEDIATE LUNAR LANDER (for TRS-80™) Your automatic landing system has malfunctioned and you are forced to make a manual landing. Taking control at 15,000 feet, the pilot's success requires precise manipulation of the lunar lander's retro-rockets. After mastering the basic game, the able commander will be taxed by the difficult intermediate program. $7.95. Order no. 0001R.

• SPACE TREK II (for the TRS-80™) This Space Trek game will challenge the most skillful and careful player. As commander of the Enterprise, you must defend the galaxy against enemy Klingon warships using your arsenal of phasers and photon torpedoes to best advantage. The Enterprise is also equipped with impulse power for moving within the quadrant, and warp power for jumping from quadrant to quadrant. $7.95. Order no. 0002R.

• ELECTRONICS I by Rick Morel (for the TRS-80™). With this group of five programs you will be able to design coils for radio circuits, calculate the component values for tuned circuits, design preamplifiers using the LM-381 IC and timer circuits, either monostable or astable using the 555 timer IC.

Coils can be designed with the use of tables or special calculators, but they are difficult to use. This package has a coil designing program which is simple to use and the computer does all the work.

There is also a program for designing tuned circuits, audio or radio frequency—fixed or variable frequency.

The LM-381 program draws the circuit diagram and gives you the component values for your needs. It will calculate any unknown values for you. The two 555 programs do the same, giving you the diagrams for either one-shot timers or astable timing systems (clocks). It will do all the calculations for you, providing times or component values.

These programs are not only helpful around any workshop or lab, they also are wonderful for teaching the fundamentals of electronics to newcomers. $7.95. Order no. 0008R.

• GOLF/CROSS-OUT by Wilford Niepraschk and David Dillehay (for the TRS-80™). Playing golf on a computer? You bet and frustrating, too. You have your choice of seven clubs—three irons, three woods, a chip and a putter. The program has 18 different fairways, each diabolically designed. You have to pick the right club and angle for hitting the ball or else you end up in the woods or a trap. The green calls for putting skill. The program has excellent graphics and you'll have a ball . . . golf ball. CROSS-OUT is the old peg puzzle, but in computer form. You have to jump pegs, removing each one jumped. The idea is to end up with but one peg, and that one in the middle hole. The program will rate your skill at the end of each try. $7.95. Order no. 0009R.

• HAM PACKAGE I by Rick Morel and D. A. Lien (for the TRS-80™). This group of eight programs (we like to give you plenty) will be of frequent help to any experimenter or ham operator.

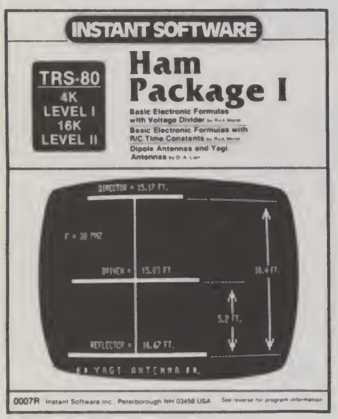
OHM'S LAW CALCULATIONS	SERIES CAPACITANCES
FREQUENCY VS REACTANCE	PARALLEL CAPACITANCES
SERIES RESISTANCES	DIPOLE DESIGN
PARALLEL RESISTANCES	YAGI DESIGN

These programs draw the circuits or antennas involved and let you fill in the pertinent data, giving you the dimensions, etc.

This series of programs is not only an excellent way to learn the fundamentals of electronics and the handling of simple formulas, they will also be of constant use in any workshop or hamshack $7.95. Order no. 0007R.

• OIL TYCOON by Dwight W. Meyer (for the TRS-80™) In this Oil Tycoon game you get to compete against a friend for the honor of being the wealthiest tycoon in the oil industry. Throughout the game you invest in research and development, explore for oil and drill with the hope of a successful strike. Not every well drilled will produce oil, and you have to rely on a geologist's report and your own instincts to come up with a gusher. There are also oil spills and blowouts to further complicate your rise to riches. $7.95. Order no. 0023R.

• BEGINNER'S BACKGAMMON/KENO —by Fred Schild (for the TRS-80™). You will never be in need of a backgammon companion again! Schild's program is designed for the newcomer to the game, but the computer's steady play will challenge the careless expert. **KENO** is a popular Las Vegas gambling game. As in many games of chance, the odds are against you, but when you win you can win big. $7.95. Order no. 0004R.



◀ Level I only

• CAVE EXPLORING/YACHT/CONCENTRATION three games by Frank Rowlett (for the TRS-80™) Cave Exploring is a game of trial, error, and exploration in which you search the inner chambers of a large cave looking for treasure while trying to avoid the dangerous creatures that guard the goods. Yacht is a two-player game based on the dice game of Yahtzee. The computer tosses the dice for you and automatically keeps score. Concentration is a computer version of the television game. Two players compete in trying to match the prizes hidden behind the twenty-five numbered squares that appear on the screen. $7.95. Order no. 0010R.

ONLY $7.95 ea.

Instant Software™ Inc

Peterborough, New Hampshire 03458 603-924-38

Ronald W. Anderson
3540 Sturbridge Ct.
Ann Arbor MI 48105

Learn BASIC— with BASIC

Computerware has a 3-part cassette course in BASIC that makes learning the language easy.

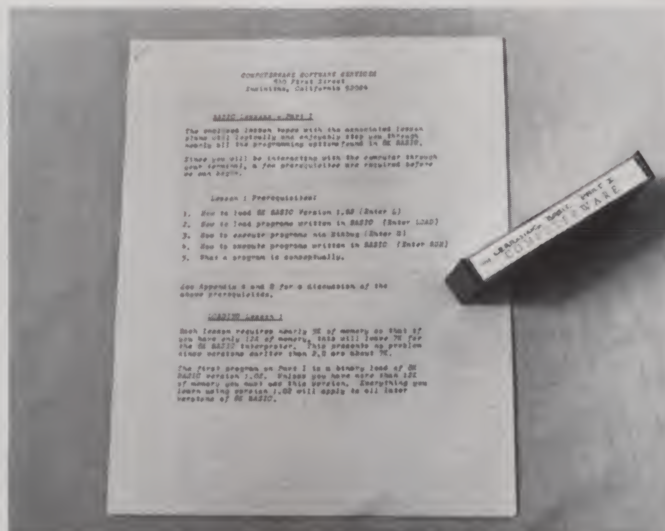
"Learn BASIC," from Computerware Software Services, is a three-part cassette course on the BASIC language. It is designed specifically around SWTP 8K BASIC version 2.0

Part 1

The idea of using a microcomputer for teaching, specifically to teach programming, intrigues me. When I received part 1, I had some difficulty loading it. I tried a second cassette player that I had handy, and had instant success. (Strangely, the two players are the same make and model.)

Each part of the course is divided into four lessons. Part 1 could be called "Basic BASIC." Lesson 1 starts with a description of program statements and the statement format (i.e., the use and meaning of line numbers, etc.). The statements or instructions PRINT, RUN and END are introduced. PRINT is discussed with regard to printing strings and evaluating numerical expressions. Next, program editing is discussed, followed by the LIST command and print formatting using the semicolon and comma, as well as the TAB command.

Lesson 2 describes the mathematical operators, use of parentheses, scientific notation, the LET statement, the concept of variables and the INPUT statement.



The Computerware cassette course.

Lesson 3 covers branching, IF THEN and GOTO, and the comparison symbols.

Lesson 4 includes the FOR-NEXT, STEP, READ and DATA statements.

Taken together, these four lessons provide an excellent introduction to BASIC. There are several example programs, and the lessons are interspersed with true/false and multiple-choice questions. Wrong answers cause the part of the lesson containing the answer to be repeated until the correct answer is given.

A good set of printed notes on the lessons with supplementary material is provided. This part of the course is useful for a computer hobbyist to teach other family members a little about BASIC. My ten-year-old, who is interested in the computer to the extent of playing games, was able to work through the first lesson without coaching in a reasonable amount of time, and the lesson held her interest throughout.

Part 2

Here, for me, is where the lessons go into the goodies of BASIC. Without repeating a lesson-by-lesson list, this part covers:

String variables
DIMension statements
Subscripted variables
PORT command
Subroutines via GOSUB and RETURN
PEEK and POKE
Applications such as tables
The ASCII interface functions LEN, ASC, CHR$, VAL
Miscellaneous functions INT, RND, DEF FN, POS

If you are a hobbyist like me, without any formal training in BASIC or other high-level languages, you will begin to see some of the power of BASIC. For example, the ASCII functions are very useful.

If you have an SWTP CT-1024 terminal and try to PRINT "(control P)";"(control V)" to home up the cursor and erase the screen, you'll find this works fine; but when you try to list the program, it homes up and clears the screen in the middle of your listing. You will find that PRINT CHR$(16); CHR$(22) will perform the same function and list perfectly.

Should you want to home up and clear on an ADM-3, use PRINT CHR$(26). You will find out how to use STR$, LEN, and TAB, and, with a little ingenuity, will be able to center a text string (even a string input by the program, of variable length).

You will also learn how to "right justify" printout of the

results of dollar calculations so that the dollars and cents line up in columns, regardless of the number of digits in the dollars amount. By using these functions, I found that I could print a result to five significant figures, regardless of the position of the decimal point.

Part 3

Believe it or not, you can, without any additional hardware, read and write data to cassette under *Program Control*. Lesson 9 tells how, and it is not difficult. Lessons 10 through 12 cover the operation of MIKBUG, including the use of some of its subroutines in your programs. This treatment is brief, but again, the notes are helpful. A little knowledge of assembly-language programming is necessary to understand this part of the lessons.

Comments

Until I received my SWTP 8K BASIC, I had not been exposed to a full BASIC. The SWTP manual adequately describes the functions and commands, but does not give one clue as to how or why they are used in a program. None of the textbooks I found at the local computer stores discuss any of the functions covered in the course.

Sure, the programs are a gimmick, but they make the learning fun and easy. The information alone is worth the price, but I do have a few negative comments.

The lessons are progressively shorter; some of the latter ones are about 30 percent shorter than the first. I found it difficult to load the cassettes, though once I found the right combination of recorder and gain setting, they loaded flawlessly. There is an error in one of the examples on scientific notation. See if you can catch it!

In part 3 I would have preferred more descriptions of the techniques, and less of the basics of MIKBUG, but then, I've been doing considerable machine-language programming and in the process have become quite familiar with MIKBUG. The part describing subroutines in MIKBUG is good, but a few examples would help there, too.

All in all, I'd say they are worth the price.*

Part 1 Understanding BASIC
Part 2 Extended BASIC
Part 3 More BASIC and MIKBUG

Computerware
830 First St.
Encinitas CA 92024■

**Editor's note (December 1978): Computerware's latest catalog lists "Learn BASIC Package" as a whole, price $39.95, available on tape or disk. Disk users should specify SWTP or Smoke Signal Broadcasting.*

John R. Ellis
150 Bry-Lynn Drive
West Melbourne FL 32901

Use Flowcharts to Communicate

The term "go with the flow" connotes conformity. However, even nonconformists should read and follow, along with the rest of us, the guidelines presented in this article.

One feature that sets man apart from the remainder of the animal kingdom is his ability to represent his thoughts in symbols. This ability constitutes one half of the process we refer to as communication.

For communication to be successful, it must include three elements: the transmitter, a set of commonly understood symbols and the receiver. The transmitter uses symbols from the set to represent his ideas. If the receiver can translate those symbols back to the original ideas, the communication will be successful. For most of our everyday communications, we select symbols (either written or spoken) from a set called the English language. In fact, I am communicating with you right now through the symbols (words) printed on this page.

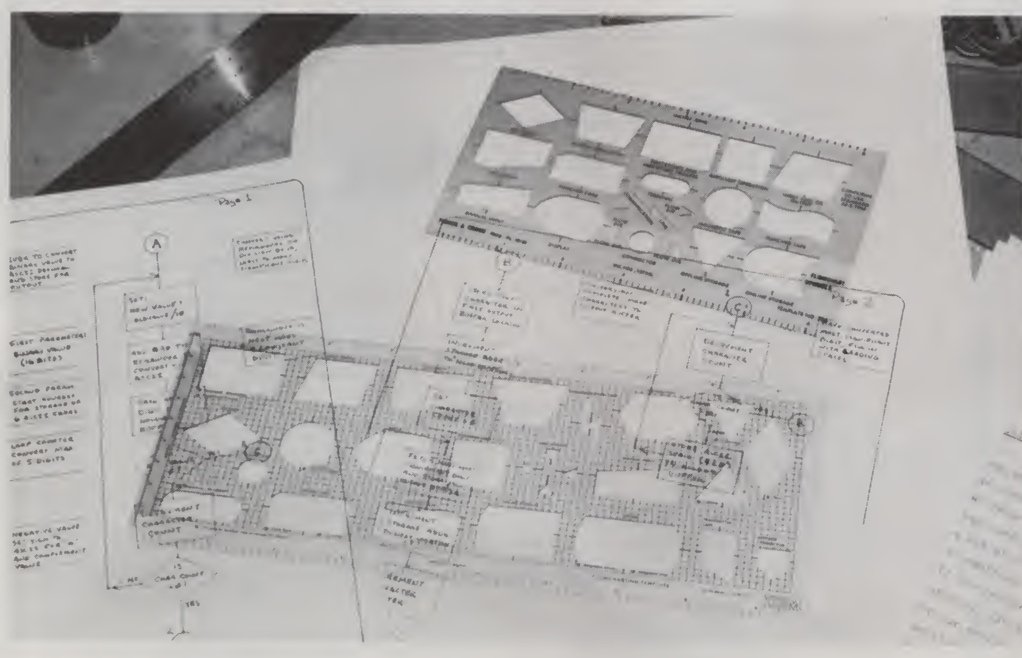
Much of the responsibility for successful communication lies with the transmitter. It is his responsibility to consider the receiver and select symbols that will be meaningful to him. You've seen this demonstrated by adults talking to young children.

Unfortunately, among adults the goal of successfully exchanging ideas is often overshadowed by the desire to impress the receiver with the transmitter's knowledge of symbols that fall outside the receiver's experience. In technical discussions, this becomes extremely easy if the receiver is not trained in the same field as the transmitter. A familiar example is the doctor who explains a patient's ailment in Latin terms. Although technically accurate, the words are meaningless unless the patient has had medical training.

An interesting extension of this occurs when the transmitter invents his own new symbols and injects them into the technical communication. The poor receiver cannot differentiate between the invented and commonly accepted symbols. A comic example of this is the car mechanic who throws a little double-talk into his description of the car's problem to justify the high repair bill because he knows the attractive young owner doesn't know the difference between a cross-ventilated aspirator and a connecting rod anyway.

On occasion, the transmitter just doesn't know the commonly accepted symbols and resorts to invention in pure desperation. The computer industry has a wide variety of symbols, many of which are finding widespread use by hobby computerists.

It appears, however, that one set of symbols in particular has not been made adequately available to hobby computerists. I refer to the *Flowchart* symbols. While reading through the back issues of some magazines, I was amazed by the ingenuity of some authors in creating new symbols or assigning new meanings to commonly defined symbols.



Two examples of flowcharting templates that conform to the ANSI standard. Note that most you find do have the ANSI symbols.

Flowcharts and the Hobby Computerist

Flowcharting is particularly important for the hobby computerist who writes for a magazine such as *Kilobaud*. The good author will recognize that readers with equipment different from his are interested in what he has done. If he wraps his whole article around his machine language or particular dialect of BASIC, he will fail to communicate with much of his interested audience.

A good representation of the program logic in flowcharts can allow implementation on most any machine. To ensure widespread understanding of his flowcharts, the author should use symbols with commonly accepted meanings. What I propose is that all *Kilobaud* authors accept a single standard, one that is widely followed in the computer industry.

The standard I refer to is the American National Standard Flowchart Symbols and Their

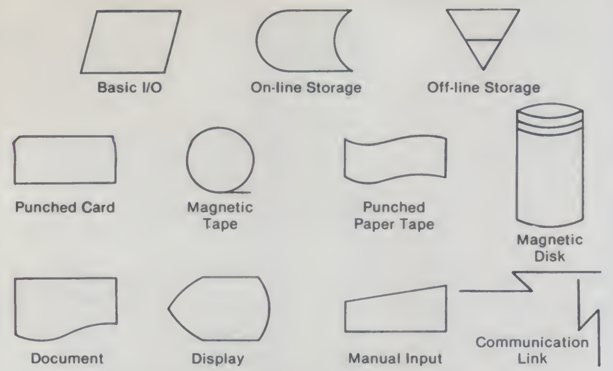


Fig. 1. Input/output function symbols.

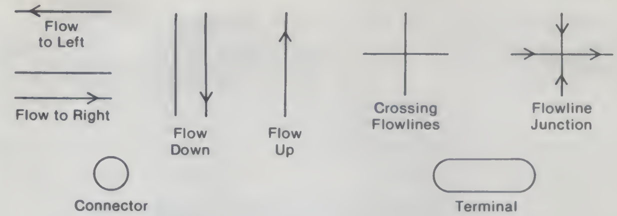


Fig. 3. Flow direction symbols.

Usage in Information Processing, X3.5-1970 (commonly referred to as ANSI X3.5). There is a variety of flowcharting templates available with these symbols, two of which are shown in the accompanying photo. The one I particularly like has an IBM logo (number GX20-8020-1), as it is cut from plastic imprinted with a cross-hatch grid that allows easy alignment.

The following is a discussion of the ANSI X3.5 standard, particularly as it applies to the hobbyist.

ANSI X3.5— General Information

Flowchart symbols are used to represent functions, or operations, and are interconnected to depict the logical relationship among those functions. The four basic functions identified in ANSI X3.5 are input/output, processing, flow direction and annotation. For each of these there is a basic symbol that can represent any function of that type. However, there are also specialized symbols commonly used in place of the basic symbol to convey additional information about the function.

The symbols are combined with a brief narrative description of the specific operation in its application. This information is usually written in the symbol.

Input/Output Symbols

Fig. 1 shows some of the input/output function symbols given in ANSI X3.5. The basic I/O symbol (top left) is used to represent any input/output operation whereby information is read into, or written out of, the computer.

The on-line storage symbol is used to represent an I/O operation that stores or retrieves data to or from a device under control of the computer. This includes a device such as a floppy disk. A cassette tape unit would qualify as an on-line storage device only if the processor could control such functions as starting, stopping and rewinding the unit under its own control. The off-line storage symbol is used to represent data storage with an off-line storage device. This includes paper tape, most cassette tape systems, etc.

I/O operations with a specific device can be represented with a symbol assigned to that device. The ones I've included are for: punched-card devices (reader or punch), magnetic-tape units (including cassettes), punched paper-tape units (reader or punch) and magnetic-disk units (including floppies). The representation of hard copy and terminals is divided between two symbols depending on the application. This includes such devices as line printers, CRT terminals, plotters or teleprinters.

If the information is displayed for use by the operator at the time of the processing (e.g., to prompt him that the program is ready for a new command, as in many programmed games), then that output is represented as being to the display symbol. If the output is for use off-line (a little difficult with a CRT!), then the document symbol should be used (for program listings, etc.). Console keypads, terminal keyboards, etc., are represented as a manual input.

When the I/O operation is over a telecommunications link, as is becoming increasingly popular with the hobbyist, based on recent articles, the communications symbol is used. Rules provided below for indicating flow direction also apply to this symbol.

Processing Symbols

The most common processing function symbols are shown in Fig. 2. The basic symbol for a "process" may be used for any processing function. It is probably the most commonly used symbol as it depicts the majority of operations that any program must perform. However, in practice it is considered bad form to use the basic symbol in place of one of the more specific symbols for decisions or subroutine calls.

The decision symbol is used to represent an operation that results in selecting one from a number of paths to be followed to the next function. Normally, this symbol has a question in it with the answers shown on the outgoing lines corresponding to the appropriate paths.

The predefined process symbol is used to represent a named process (such as a subroutine) that is defined elsewhere. Execution of this process is generally a subroutine call (transfer of control to the named procedure) with the outgoing line indicating the return point. The auxiliary operation symbol represents a function that is performed off-line with equipment not under direct control of the processor.

The remaining group of processing symbols are particular-

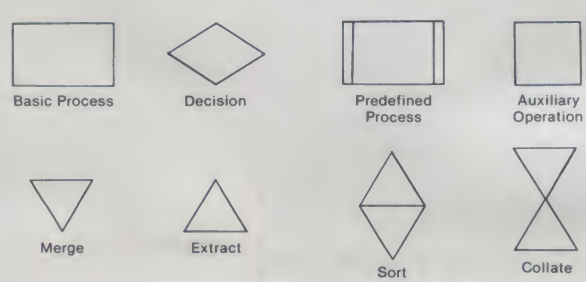


Fig. 2. Processing function symbols.

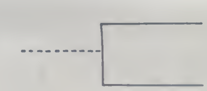


Fig. 4. Annotation symbol.

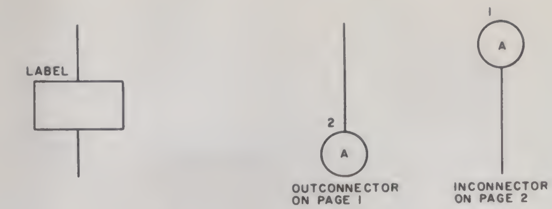


Fig. 5. Cross-reference Labels.

ly useful in the design of business or word-processing systems. They apply to the operations related to data sets (files, etc.): merge two sets, extract one set from another, sort a set or collate two or more sets.

Flow Direction Symbols

Flow is represented by lines drawn between symbols. Normal flow is from left to right and from top to bottom. When these basic directions are violated, an arrowhead must be used to show the correct flow direction. The arrowhead may also be used with the normal flow direction lines to increase clarity. Fig. 3 illustrates these legal flowlines.

Flowlines may cross when there is no interrelation between the flow paths. Two or more incoming flowlines may join to a single outgoing flowline. This is referred to as a junction. Every flowline entering or leaving a junction should incorporate an arrowhead near the junction. Examples of crossing flowlines and a junction of flowlines are also shown in Fig. 3.

When it is not practical to draw a line directly between two symbols because of possible confusion from using long lines or due to the lack of room on a single page, a pair of connector symbols may be used to represent the continuity of flow. Flow will be shown as entering the first connector symbol, which will have a reference identifier. Flow will continue out of the second connector symbol with the same identifier. This can also be implemented as a junction where several flows enter a common point.

The terminal symbol is used to represent a terminal point in the flowchart, such as the start, the end, an interrupt or a halt. When used to indicate the start point, the program or subroutine name is often given in the terminal symbol.

Annotation Symbol

A comment, or further explanation of a function (such as describing the operation to be performed by a subroutine call), can be added to the flowchart using the annotation symbol shown in Fig. 4. The broken line runs to the appropriate symbol in the flowchart. Experience has shown that liberal use of this symbol greatly enhances the understanding of the flowchart.

Cross-Reference Labels

It often becomes necessary

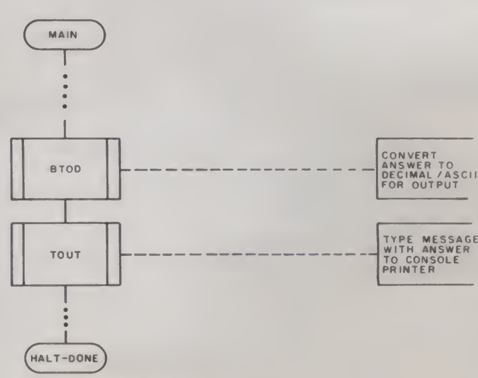


Fig. 6. Mainline-calling sequence for example.

Reader Service—see page 179

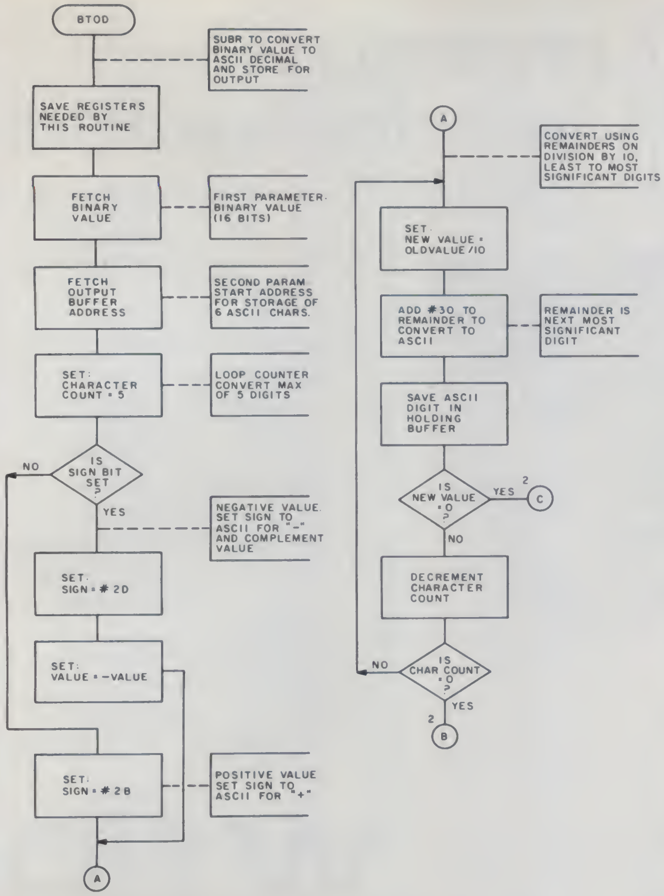


Fig. 7a. Example of flowchart for subroutine to convert a binary number to ASCII for decimal equivalent.

to relate a point in a program listing to the corresponding point in a flowchart. Placing any symbolic labels used by the program (mnemonic labels from assembly language or BASIC line numbers) to the upper left of the function symbol, as shown in Fig. 5, at that corresponding point in the program flowchart allows easy cross-references between these two documents.

Interconnector symbols can create a problem when they lead off the current page, and many pages are involved in searching for the continuation point. It would be handy if the connector told you where to look for its counterpart. The example shown in Fig. 5 illustrates the proper procedure. The page of the paired connector is given above and to the left of the connector.

Style

Although these rules may appear quite rigid, they are susceptible to individual programmers' styles. The number of arrowheads (only when required versus on every flowline), the use of connectors on a single page in place of longer flowlines, the number of annotations used, etc., will reflect the individual programmer's idea of what he thinks looks good or best conveys his meanings.

Another practice that I have found helpful for both coding a program (to ensure all paths are covered) and understanding a program already coded is to run all symbols down single-line columns that correspond with the order in the program code. Again, this is matter of the programmer's preference as ANSI X3.5 does not address page layout.

An Example

Figs. 6 and 7 illustrate flowcharts drawn by the ANSI X3.5 standards. (Note that "#" preceding a number indicates that

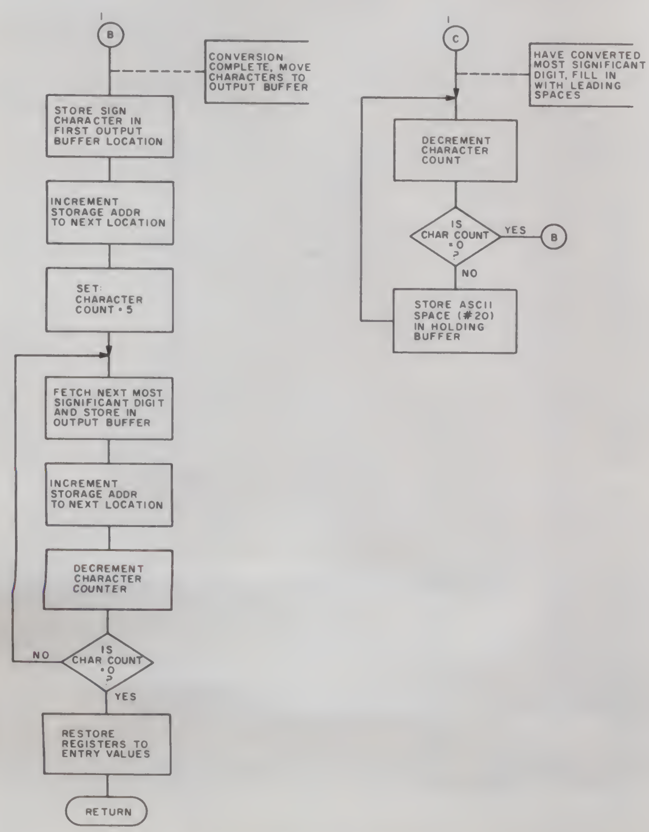


Fig. 7b. Continuation of example flowchart.

the value is hexadecimal.) The mainline sequence shown in Fig. 6 includes a call to the subroutine BTOD. This subroutine is depicted in Fig. 7. It will convert a binary value to up to five digits of the decimal equivalent. The decimal digits are stored as ASCII characters in memory locations specified by the calling routine, presumably an output buffer.

The logic described in this example could be applied to any machine. When the actual coding takes place, some minor changes to the flowchart, such as adding program labels to the appropriate points in the diagram, could be made.

Summary

As computer hobbyists adopt the techniques employed by their professional counterparts, I hope to see the acceptance of good flowcharting to convey information about a program's logic in a machine-independent manner. I consider "good" flowcharts as those that are easily understood by all who might look at them. Certainly those to be published should conform to a generally accepted standard, such as ANSI X3.5.

I have not attempted to cover the entire standard. Instead, I have discussed those points that most probably apply to programs developed by *Kilobaud* readers. The complete standard can be consulted for more information, if necessary. However, after ten years in the field, I cannot recall using symbols not covered here.

One way to encourage acceptance of this standard would be for computer journals to require compliance for all material they print. This would also put the proper techniques on display for beginners to pick up—just as a child learns to speak: by constant exposure! With everyone using the same symbols, receivers (readers) will be better assured of good communications from the transmitters (writers). ■

Reader Service—see page 179

DO YOU WANT AN **Intelligent E-Prom** PROGRAMMER

WITH MULTIPLE PERSONALITIES FOR UNDER **$200.00?**

THEN BE **Smart** AND PLUG OUR **"Dumb"** PROGRAMMER

INTO YOUR COMPUTER!

PROGRAM 2708, 2758, TMS2716, TMS2516, INT2716+5V, TMS2532, INT2532.
REQUIRES TWO PARALLEL I/O PORTS.
USES ZERO INSERTION FORCE SOCKET.
SELF CONTAINED-NO ADDITIONAL POWER SUPPLIES NEEDED.

E-PROM PROGRAMMER PRICE-$179.95

SOFTWARE PRICE-$19.95
(PLEASE SPECIFY COMPUTER)

TRS-80

EXPANSION MOTHERBOARD	PRICE-$99.95
I/O BOARD	PRICE-$79.95
E-PROM BOARD	PRICE-$49.95
ASSEMBLER/MONITOR	PRICE-$29.95
HARDCOPY INTERFACE 20MA. CURRENT LOOP	PRICE-$59.95
RS-232C INTERFACE	PRICE-$74.95

ALL OF OUR PRODUCTS ARE ASSEMBLED AND TESTED!

PET

FIRMWARE I/O BOARD INCLUDES TWO E-PROM SOCKETS, 2708 OR 2716, TWO I/O PORTS, ONE 6522 V.I.A.	PRICE-$99.95
HARDCOPY INTERFACE 20MA. CURRENT LOOP	PRICE-$59.95
RS-232C INTERFACE	PRICE-$74.95
ASSEMBLER/MONITOR	PRICE-$29.95
CABLES FOR E-PROM PROGRAMMER	PRICE-$9.95

GPA ELECTRONICS ✓G24

GPA ELECTRONICS
906 BLAIR AVE.
OAKLAND CA 94611
(415)654-3898

VISA

master charge

IF YOU OWN A TRS 80 *.......

TRS-80 is a product of Radio Shack, A Tandy Corporation

SELECTRA-PRINT PUTS YOU JUST A CABLE LENGTH AWAY FROM A "HARD—COPY" (Typewriter Quality: clean, clear, high-fidelity) Word Processing System!

. . .just a simple hook-up with the supplied cables and your *SELECTRA-PRINT* is ready to run.

For Word Processing on a TRS-80, for example, just command "LPRINT" and *SELECTRA-PRINT* automatically outputs clear, clean, high-fidelity, hard copy. . . . and of course you can use it to print-out any other information you might need.

SPECIAL NOTE:
SELECTRA-PRINT is a Selectric II typewriter and although it has been modified for computer print-out, it may still be used as a standard office typewriter.

PRICE: $1,850.00* *TRS-80 Version $1925.00

OPTIONS:

Dual Pitch	$125.00
Correction Feature	$125.00
Noise Reduction Feature	$50.00
RS-232 Interface	$195.00

Direct International Sales:
International Sales Division
17648 Orna Drive
Granada Hills, CA 91344, USA

Discounts Available to
EDUCATIONAL ACCOUNTS
Contact: **Delores Sun P.O. Box 8394**
***Ann Arbor, MI 48105* (313) 665-8514**

***** ***SELECTRA-PRINT is VERSATILE!***
SELECTRA-PRINT is compatible with most micro- and mini-computers including *PET* ● *Apple* ● *Heath H8* ● *IMSAI* ● *Cromemco* ● *Alpha Micro-systems* ● *Space Byte* ● *North Star Horizon* ● *SWTP* ● *Vector Graphic* ● *Sol* ● *Polymorphic* ● *Digital Group* ● *Ohio Scientific* ● *Altair* ● *Sorcerer* *Xitan* ● *Rex* ● *KIM* ● *EXORcisor* . . .

"From now on, PRINT-OUT takes on a new meaning"

M30

micro computer devices inc.

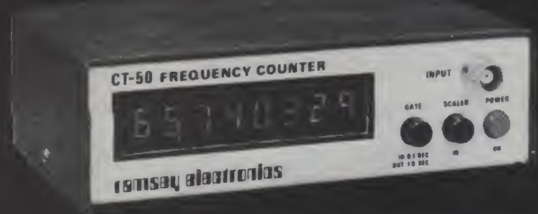
mcd

Dept. K1

960 E. Orangethorpe, Bldg. F
Anaheim, California 92801
Telephone (714) 992-2270

"Innovators to the Microcomputer Industry"

✓ Reader Service—see page 179



FREQUENCY COUNTER KIT

Outstanding Performance

Incredible Price

$89.95

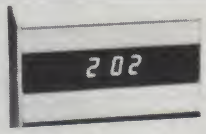
CT-50

The CT-50 is a versatile and precision frequency counter which will measure frequencies to 60 mHz and up to 600 mHz with the CT-600 option Large Scale Integration, CMOS circuitry and solid state display technology have enabled this counter to match performance found in units selling for over three times as much. Low power consumption (typically 300-400 ma) makes the CT-50 ideal for portable battery operation Features of the CT-50 include large 8 digit LED display, RF shielded all metal case, easy pushbutton operation, automatic decimal point, fully socketed IC chips and input protection to 50 volts to insure against accidental burnout or overload And, the best feature of all is the easy assembly. Clear, step by step instructions guide you to a finished unit you can rely on **Order your today!**

SPECIFICATIONS:
Frequency range 6 Hz to 65 mHz 600 mHz with CT-600
Resolution: 10 Hz @ 0.1 sec gate, 1 Hz @ 1 sec gate
Readout 8 digit, 0.4" high LED, direct readout in mHz
Accuracy adjustable to 0.5 ppm
Stability 2.0 ppm over 10 to 40 C, temperature compensated
Input BNC, 1 megohm 20 pf direct, 50 ohm with CT-600
Overload: 50VAC maximum, all modes
Sensitivity less than 25 mv to 65 mHz, 50-150 mv to 600 mHz
Power 110 VAC 5 Watts or 12 VDC @ 400 ma
Size 6" x 4" x 2", high quality aluminum case, 2 lbs
ICS 13 units, all socketed

CT-50, 60 mHz counter kit	$89.95
CT-50WT, 60 mHz counter, wired and tested	159.95
CT-600, 600 mHz scaler option, add	29.95
CB-1 Color TV calibrator-stabilizer	$14.95
DP-1 DC probe, general purpose probe	12.95
HP-1 High impedance probe, non-loadir	15.95

CAR CLOCK



The UN-KIT, only 5 solder connections

Here's a super looking, rugged and accurate auto clock, which is a snap to build and install. Clock movement is completely assembled—you only solder 3 wires and 2 switches, takes about 15 minutes! Display is bright green with automatic brightness control photocell—assures you of a highly readable display, day or night Comes in a satin finish anodized aluminum case which can be attached 5 different ways using 2 sided tape Choice of silver, black or gold case (specify)

DC-3 kit, 12 hour format	$22.95
DC-3 wired and tested	$29.95
110V AC adapter	$5.95

OP-AMP SPECIAL

741 mini dip	12/$2.00
B1-FET mini dip, 741 type	10/$2.00

FM MINI MIKE KIT

A super high performance FM wireless mike kit! Transmits a stable signal up to 300 yards with exceptional audio quality by means of its built in electret mike Kit includes case, mike, on-off switch, antenna, battery and super instructions This is the finest unit available

FM-3 kit	$12.95
FM-3 wired and tested	16.95

VIDEO TERMINAL

A completely self-contained, stand alone video terminal card. Requires only an ASCII keyboard and TV set to become a complete terminal unit. Two units available, common features are: single 5V supply, XTAL controlled sync and baud rates (to 9600), complete computer and keyboard control of cursor. Parity error control and display. Accepts and generates serial ASCII plus parallel keyboard input. The 3216 is 32 char. by 16 lines, 2 pages with memory dump feature. The 6416 is 64 char. by 16 lines, with scrolling, upper and lower case (optional) and has RS-232 and 20ma loop interfaces on board. Kits include sockets and complete documentation.

RE 3216, terminal card	$149.95
RE 6416, terminal card	189.95
Lower Case option, 6416 only	13.95
Power Supply Kit	14.95
Video/RF Modulator, VD-1	6.95
Assembled, tested units, add	60.00

CLOCK KITS

our Best Seller your Best Deal

Try your hand at building the finest looking clock on the market Its satin finish anodized aluminum case looks great anywhere, while six 4" LED digits provide a highly readable display This is a complete kit, no extras needed, and it only takes 1-2 hours to assemble Your choice of case colors, silver, gold, bronze, black, blue (specify)

Clock kit, 12/24 hour, DC-5	$22.95
Clock with 10 min ID timer, 12/24 hour, DC-10	27.95
Alarm clock, 12 hour only, DC-8	24.95
12V DC car clock, DC-7	27.95

For wired and tested clocks add $10 00 to kit price

Under dash car clock

12/24 hour clock in a beautiful plastic case features 6 jumbo RED LEDS, high accuracy (1min/mo), easy 3 wire hookup, display blanks with ignition, and super instructions Optional dimmer automatically adjusts display to ambient light level

DC-11 clock with mtg bracket	$27.95
DM-1 dimmer adapter	2.50

CALENDAR ALARM CLOCK

The clock that's got it all: 6- 5" LEDs, 12/24 hour, snooze, 24 hour alarm, 4 year calendar, battery backup, and lots more The super 7001 chip is used. Size 5x4x2 inches
Complete kit, less case (not available)

DC-9	$34.95

PRESCALER

Extend the range of your counter to 600 mHz Works with any counter Includes 2 transistor pre-amp to give super sens, typically 20 mv at 150 mHz Specify ÷ 10 or ÷ 100 ratio.

PS-1B, 600 mHz prescaler	$59.95
PS-1BK, 600 mHz prescaler kit	49.95

30 Watt 2 mtr PWR AMP

Simple Class C power amp features 8 times power gain. 1 W in for 8 out, 2 in for 15 out, 4 W in for 30 out Max. output of 35 W, incredible value, complete with all parts, less case and T-R relay

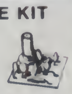
PA-1, 30 W pwr amp kit	$22.95
TR-1, RF sensed T-R relay kit	6.95

Hard to find PARTS

LINEAR ICs		REGULATORS	
301	$ 35	78MG	$1 25
324	1 50	723	50
380	1.25	309K	85
380-8	75	7805	85
555	45	78L05	25
556	85	7905	1 25
566	1 15	7812	85
567	1 25	7912	1 25
1458	50	7815	85
3900	50	**TTL ICs**	
CMOS ICs		74S00	35
4011	20	7447	65
4013	35	7475	50
4046	1 85	7490	50
4049	40	74196TI	1 35
4518	1 25	**SPECIAL ICs**	
5369	1 75	11C90	13 50
TRANSISTORS		10116	1 25
2N3904 type	10/1 00	4511	2 00
2N3906 type	10/1.00	5314	2 95
NPN 30W Pwr	3/1 00	5375AB	2 95
PNP 30W Pwr	3/1.00	7001	6.50
2N3055	.60	4059 ÷ N	9 00
UJT 2N2646 type	3/2 00	7208	17 95
FET MPF102 type	3/2 00	**LEDs**	
UHF 2N5179 type	3/2.00	Jumbo red	8/1.00
MRF-238 RF	11 95	Jumbo green	6/1.00
SOCKETS		Jumbo yellow	6/1.00
8 pin	10/2 00	Mini red	8/1.00
14 pin	10/2.00	Micro red	8/1.00
16 pin	10/2.00	BiPolar	75
24 pin	4/2.00	**FERRITE BEADS**	
28 pin	4/2.00	With info, specs	15/1.00
40 pin	3/2.00	6 hole balun	5/1.00

Ramsey's famous MINI-KITS

FM WIRELESS MIKE KIT

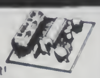


Transmits up to 300' to any FM broadcast radio, uses any type of mike Runs on 3 to 9V Type FM-2 has added sensitive mike preamp stage
FM-1 kit **$2.95** FM-2 kit **$4.95**

VIDEO MODULATOR KIT

Converts any TV to video monitor Super stable, tunable over ch 4-6 Runs on 5-15V, accepts std video signal Best unit on the market!
Complete kit, VD-1 **$6.95**

SUPER SLEUTH



A super sensitive amplifier which will pick up a pin drop at 15 feet! Great for monitoring baby's room or as general purpose amplifier Full 2 W rms output, runs on 6 to 15 volts, uses 8-45 ohm speaker
Complete kit, BN-9 **$5.95**

COLOR ORGAN/MUSIC LIGHTS

See music come alive! 3 different lights flicker with music One light for lows, one for the mid-range and one for the highs Each channel individually adjustable, and drives up to 300W Great for parties, band music, nite clubs and more
Complete kit, ML-1 **$7.95**

TONE DECODER

A complete tone decoder on a single PC board Features 400-5000 Hz adjustable range via 20 turn pot, voltage regulation, 567 IC. Useful for touch-tone decoding, tone burst detection, FSK, etc Can also be used as a stable tone encoder Runs on 5 to 12 volts
Complete kit, TD-1 **$5.95**

POWER SUPPLY KIT

Complete triple regulated power supply provides variable 6 to 18 volts at 200 ma and + 5V at 1 Amp Excellent load regulation, good filtering and small size Less transformers, requires 6.3V @ 1 A and 24 VCT
Complete kit, PS-3LT **$6.95**

LED BLINKY KIT

A great attention getter which alternately flashes 2 jumbo LEDs Use for name badges, buttons, warning panel lights, anything! Runs on 3 to 15 volts
Complete kit, BL-1 **$2.95**

WHISPER LIGHT KIT

An interesting kit, small mike picks up sounds and converts them to light The louder the sound the brighter the light Completely self-contained, includes mike, runs on 110VAC, controls up to 300 watts
Complete kit, WL-1 **$6.95**

SIREN KIT

Produces upward and downward wail characteristic of a police siren 5 W peak audio output, runs on 3-15 volts, uses 3-45 ohm speaker
Complete kit, SM-3 **$2.95**

ramsey electronics
BOX 4072, ROCHESTER, N.Y. 14610

PHONE ORDERS CALL
(716) 271-6487

TERMS: *Satisfaction guaranteed or money refunded* COD, add $1.50. Minimum order, $6 00 Orders under $10.00, add $.75 Add 5% for postage, insurance, handling Overseas, add 15% NY residents, add 7% tax

RONDURE COMPANY

2522 BUTLER ST.
DALLAS, TEXAS 75235
R7 214—630-4621

the computer room

ASCII SELECTRIC

SPECIAL SALE
$875.00

Used Working and Clean

TESTED WITH NEW ASCII ELECTRONICS
Factory Reconditioned
$1075.00

Printer Mechanism: Heavy duty input/output, Series 745.

Weight: 120 lbs. Dimensions: 29"Hx35"Wx33"D.

Print Speed: (14.8 characters per second)

Platen: 15" wide, pin feed or form feed device optional (132 print positions).

Parallel output only—15 characters per second accepts 7 bit ASCII parallel w/strobe & prints on Selectric. The unit still works as a typewriter in off-line mode.

ASCII Selectric with ASCII parallel electronics. Immediate Delivery—Shipped from inventory.

DATEL SELECTRIC (IBM Selectric Mechanism)

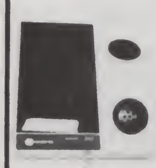
"As Is" Complete	$395.00
Working	$495.00
Reconditioned	$695.00

Specifications:
- Size: 21"Wx21"Dx8"H.
- Power Input 115 Volt Hz
- inferace: RS232
- Weight: 54 lbs. (Shipping weight 65 lbs.)
- 15" Carriage
- 15 CPS
- Correspondence code
- Half Duplex
- 132 Print Positions, 10 Pitch

Used — $395

NOVATION DC3102A

Used Working



$150.00

RS232 Connection
300 Baud

TI 990/4

Single Board 16 Bit Micro Computer

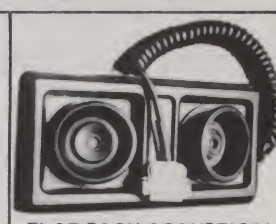
NEW $250.00

USED MODEMS & COUPLERS

NOVATION TC102 (Acoustic)	$45.00
AJ 233 (Acoustic)	$25.00
CF 318 (Hard Wire)	$25.00

SHUGART MINI-FLOPPY DRIVE

NEW PRICE
$325.00 ea.
Model SA-400



FLAT PACK ACOUSTICAL MODEM PICK-UP

Useable with most modem chips/kits
Used — $17.50 (w/prints)

ORDERING INFORMATION:
We ship the same day we receive a certified check or money order. Texas residents add 5% sales tax. Please call if you have a question. Write for our CATALOG of many parts, terminals, printers, etc. All items subject to availability. Your money returned if we are out of stock.

SHIPPING INFORMATION:
Modems: $2.00 each; 2 for $4.00 UPS.
Large Items & Parts: Specify Freight or Air Freight Collect
Foreign Orders: Add appropriate freight or postage.
We now take Master Charge and Visa orders, Specify full number, bank number and expiration date.

ATTENTION TRS-80 & APPLE USERS

A PRINTER FOR YOUR COMPUTER

TERMS: VISA, MASTERCHARGE, Cashier Check or Money Order. C.O.D. with 10% down. Shipping Via Air or Truck collect.

3 S SALES S55
P.O. BOX 45944
TULSA, OK 74145
918/622-1058

$995.00
MODEL 3S-80 for TRS-80
Ready to plug into your expansion interface.

MODEL 3S-PP
for computers with 8 bit serial port.

MODEL 3S-SS
for computers with RS-232 port.

$1095.00 for
MODEL 3S-AA
Includes RS-232 card for AppleII
Specify model number on order.

- Ready to plug into your computer
- Very high quality print
- Completely refurbished IBM 731 I/O Selectric terminal in a new table
- Upper & lower case removable type ball
- Special I/O interface
- Heavy duty re-mfg. IBM power supply

SPINTERM T.M.

FASTER THAN THE DEVIL! at 55 characters per second.

More characters—up to 128 on an interchangeable print thimble.

Longer life with a print thimble that gives over 30,000,000 impressions.

Quieter at 60 dB with hinged covers on, and 67 dB with covers off.

We started with the revolutionary NEC Spinwriter*, added proportional spacing, and BEHOLD! . . . a complete input/output word processing printer!

All this, plus many standard features enable the SPINTERM to outperform other printers in this range of capabilities.

Ribbon and thimble changes are fast and easy.

For instance, the forms length control provides 99 choices of lines. You can select 6 or 8 lines per inch. You have 6 baud rates at 110, 150, 200, 300, 600 and 1200. Industry standard RS-232 interfaces contain a built-in self test. The quality of print can't be beat for it's uniform impression, even through an original and 5 copies. The numeric keypad is standard, along with full ASCII keyboard. You have a choice of ribbon styles, each in a snap-on cartridge. And we can supply an optional full word processing software package.

We've barely scratched the surface of outstanding features which are standard on the SPINTERM. So write us, or call your dealer for complete details.

INPUT/OUTPUT UNLIMITED, INC.
13762 Victory Boulevard
Van Nuys, California 91401 ✓130
(213) 997-7791

SPINTERM

*Trademark of NEC Information Systems, Inc.

✓ Reader Service—see page 179

John B. Palmer
Box 23
Boonville CA 95415

A Joystick Interface for Your Altair

A joystick can add a new dimension to the interactive games you play with your Altair.

One of my reasons for buying a video display module for my Altair 8800B was to play video graphic games. After a few minor problems, I wrote a version of Pong in 8K BASIC.

The ninth grade class loved it until they found out that a keyboard is a poor substitute for a joystick. Students were complaining, "How do you make it go up? Hey, it won't stop now! How come my side won't work when he holds down his key?" I tried to explain that you must not hold down a key during the other person's turn. Then somebody said, "I got one at home that works better than that."

So, in the eyes of the ninth graders, my wonderful microcomputer was no better than a mere video game because it didn't have joysticks. That did it . . . I wanted a good, cheap joystick interface right away. I ordered a pair of joysticks from one of the mail-order firms, and they arrived in good shape and looked very good.

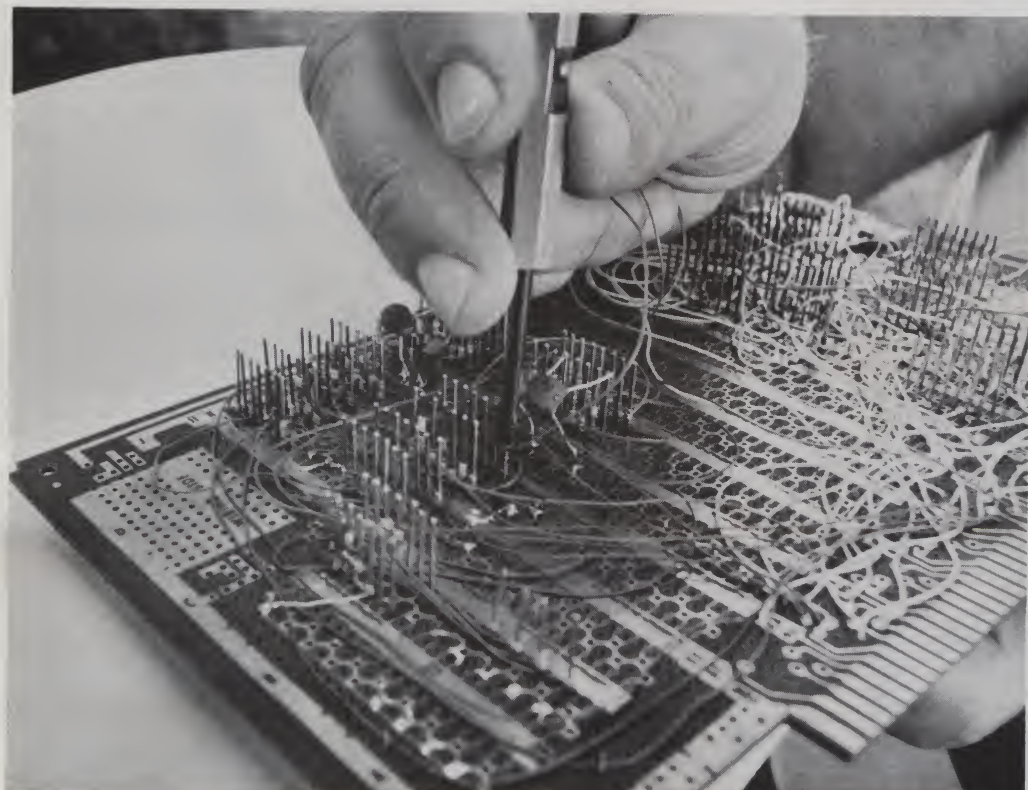
The VOM showed that the potentiometers were about 100k Ohms each. There was a "dead spot" in the middle and at either end, so the action of each joypot was not going to be very smooth. These units were intended for low-cost four-channel audio devices. Still, the price was right.

I went ahead and built a low-cost interface. The end result is a system that is more versatile than most home video games.

The circuit in Fig. 1 requires six integrated circuits and about 32 bytes of software support. I wrote a BASIC game that reads the joysticks by the USR function.

The fact that I enjoy building things was not my motivation for a home-brew design. It was really a question of cost. You can now buy a video for under $30, yet a joystick interface for the Altair 100-pin bus is about $200! The big expense is the precision analog-to-digital conversion. That is fine for commercial use; for my needs, it was just out of sight.

A study of recent magazine articles was not very encouraging, until I ran upon an article by Stephen Wozniak of the Apple Computer Company. He has the right idea. Precise voltage-to-binary conversion is not really needed for a video game. He shows how the 555 timer will convert a variable resistance, the joypot, into a positive pulse that will vary in duration in proportion to the resistance. A 555



Wire-wrapping the Vector 8800 board. *(Photos by John Yawn)*

timer can give a pulse as short as a microsecond or as long as several seconds. Being a high-impedance device, it works well with the 100k pots. The 555 is TTL compatible both for input and output.

Of course, you cannot just tie four 555s to the data bus. You will need some kind of interface. Not wishing to tie up the few I/O cards I have, I chose to build a complete interface for the timers with provision to expand later. It is very simple, has no frills and works quite well.

Fig. 1 shows a 74125 Tri-state bus driver. It becomes active when pins 1, 4, 10 and 13 are held in the low state (almost zero volts). Otherwise, it is in the high-impedance state, which means that it acts as though it isn't there.

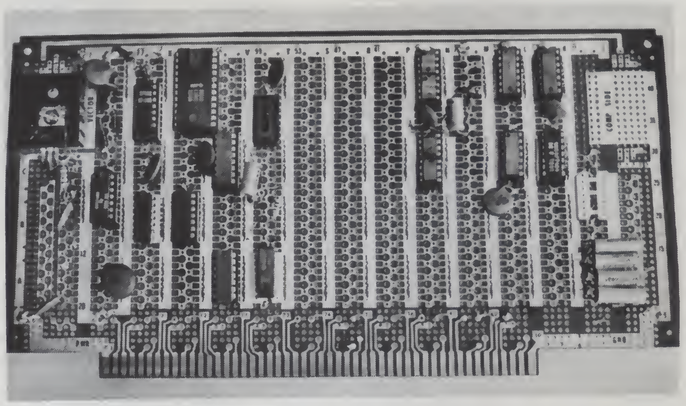
Other good devices that might be used for this include the 8T97 and the 24-pin 8212. Both are more expensive. A very good alternative is the 74LS367, now selling for under $1. It has six gates instead of the four of the 74125. The pin connections are different; only four of the eight Data In lines are used in the basic design.

Use either four 555 timers or two 565 dual timers. Again, note the different pin-outs. I put two 555s into one 16-pin wire-wrap socket. So the way I see it, the two 555s count only as one IC. That is why I said only six ICs. No other interface or I/O board is needed. The parts list must include a perfboard or Vector board, about a half-dozen resistors and maybe ten .1 uF capacitors. Part values are not critical.

Software takes the place of some of the hardware. About 32 bytes of 8080 machine code accomplish the rest of the analog-to-binary conversion. If a byte of memory is worth three cents, then I have used 96 cents of memory. Doing the whole conversion in hardware alone would cost many dollars. In view of my present capital reserve for new computer purchases, it was not hard to make a choice.

The circuit is on a Vector 8800 wire-wrap board. If you have to buy one just for this project, be prepared to part with $20. I already had one because my last project left plenty of room for another circuit.

I recommend the wire-wrap board over perfboard or etched board. A perfboard circuit can be very unstable electrically unless you use plenty of bypass capacitors and heavy bus wire for both ground and 5 volt power supply connections.



The finished joystick interface is on the right side of the board (a home-brew parallel input port is on the left).

Etched circuit overcomes the stability problem but has the drawback of being rather permanent. In case of a mistake it is much harder to unsolder than to unwrap.

By the way, if you do buy a Vector board, read the instructions the first time you try it. You may forget that the pin numbers are mirror-image reverse on the back side of the board. Here are some things you might do if you don't pay attention to Vector's detailed suggestions.

1. Wire several sockets backwards.
2. Short out the 7805 you installed on the back side.
3. Vaporize shorted traces on the motherboard!
4. Wipe out almost everything!

I will not admit which of the above actually happened to me. Don't let this stop you if you are new at home construction. Just be careful.

Software Theory

After I was sure that the circuit was wired properly, I tried a simple test program. Starting at location 300 octal, I toggled in the program listed in Program A. I used octal notation because I make fewer mistakes in translating octal to binary than hex to binary.

A sense switch is used to select which of the four analog inputs is to be counted. So I put up A8, and the data lights (on the front panel) then display a binary count that goes up or down with the vertical position to the right-hand joystick. This is only a test program.

Before I have moved the joystick to its full vertical position, the data lights have gone past 255. An overflow has occurred in register B. No *matter. A bet*ter software routine will take care of this. Right now, I know that the joystick is working.

Next, try the other three inputs—put either A9, A10 or A11 up. I find two are not working. A recheck of the pots shows I reversed a couple of wires.

The next step is a BASIC program (Program B) and a USR routine in machine code to play a video game. BASIC reads the DATA and POKEs it into consecutive memory locations at

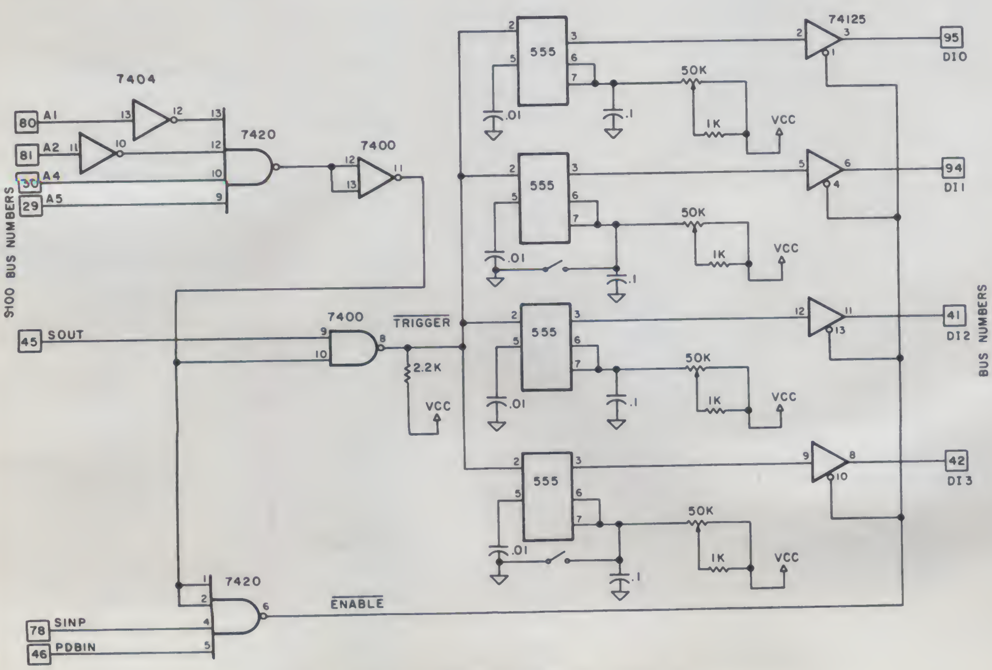
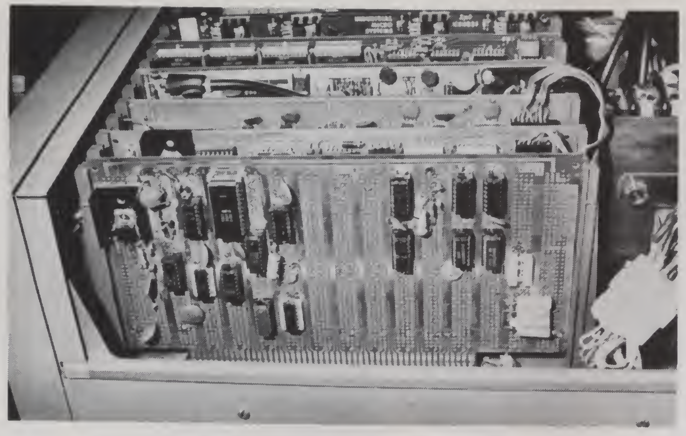


Fig. 1. Joystick interface logic diagram.



Finished board installed in Altair 8080B. White DIP socket is to connect joybox.

the top of my 16,383 byte memory. Finally, a patch is made into the place deep within BASIC where the USR function is called.

The rest of the BASIC listing, line 1000 onward, is a game that I call "Star Chase." Positions for the Enterprise, represented by *, and the Klingon, represented by #, are calculated from values returned by the joystick routine and the 555 timers.

I have Mits 8K Altair BASIC revision 4.0, which returns the joystick values in registers A and B with "hi-byte" in A, "lo-byte" in B. Registers A and B are used as a pair. Now, you and I know that A and B are not a register pair in the 8080, but this version of BASIC doesn't know that! Check your BASIC user's manual.

Not only are there variations between brand X BASIC and brand Y, but also the same supplier may make radical changes in the internal structure of his BASIC next week! So before you attempt to use this program, check your user's manual carefully.

Lacking a BASIC with the PLOT feature, I wrote the game for the VDM-1 video display by using the POKE and PEEK features. Normally the VDM-1 is located at CC00 through CFFF. To POKE and PEEK into that part of memory I had to use negative numbers.

The program first prints the rules, then clears all variables and assigns values to the most frequently used variables first. This is to reduce lookup time. Each character is put onto the screen at locations computed

```
Split-Octal   Data   Instruction
000-300       333    IN    377    ;get bit from sense sw.
    301       377
    302       137    MOV E,A      ;put it in reg E.
    303       006    MVI  B       ;zero reg B.
    304       000
    305       323    OUT  60      ;out to port 48
    306       060                 ;will trigger timers
    307       004    INR  B       ;add one to B.
    310       333    IN   60      ;input from port 48
    311       060                 ;to get timer status.
    312       243    ANA E        ;logically AND A with E.
    313       302    JNZ          ;jump if not zero. timer
    314       307                 ;which corresponds to the
    315       000                 ;bit in E has not reset.
    316       170    MOV A,B      ;ready to display.
    317       323    OUT 377      ;display count on front
    320       377                 ;panel of 8800B.
    321       303    JMP          ;go back and do it all
    322       300                 ;over again from start.
    323       000
```

Program A. Joystick interface test program.

Joystick detail.

by adding the value from the joystick to the starting location of the VDM-1.

A check is made to see if the spaceship has changed position. If so, its previous location must be an ASCII blank to simulate motion. If a spacecraft tries to occupy the same space as a binary star, then the star pair is destroyed. Whenever the Enterprise figure and the Klingon figure are in the same place at the same time, the game ends.

To make the game more interesting, "black holes" will cause either figure to vanish for a few seconds only to reappear somewhere else. In the rules, "Spock" gives his logical explanation for this.

Now, let's examine how the machine-language routine works on either the 8080 or Z-80. The general idea can also be used with the 6800 or 6502 if you have a good imagination. USR calls the routine placed in the top of the memory. It, in turn, calls a routine deep inside of BASIC that returns a 16-bit signed integer. Next, an OUT instruction sends all zeros to port 48, which is the joystick interface. This sets all of the timers. The time formula for any one of the timers to reset is: Time = 1.1 * capacity * resistance; time is microseconds, capacity is microfarads.

The program keeps increas ing the value in register B unti either the desired timer resets or register B overflows. The routine returns to the calle after first calling a program that will store the values o registers A and B.

How does the program tel which timer has reset? The value in the E register is a "mask" that makes the selec tion. For example, if timer 3 is desired, then E must have the value of four (i.e., only bit two is high). Register D has a value to scale down the count. This car be used for various functions such as correcting variations ir the time constant of the timers

You might think it easier to just change the values of the capacitors. You can spend a lot of time trying to find the value that works best for each joypot.

Line 1410 of the BASIC program "Star Chase" has constants used for the vertical position of the spaceships. Why is E1 given the value of 9729? Divide 9729 by 256 and you have 38 with a remainder of 1. So, when BASIC executes the expression USR(E1), the D register has a value of 38 and the E register has a value of 1. When K4 is used instead, the values are 38 and 4, respectively. The VDM-1 board has only 16 verti-

cal positions but has 64 horizontal. A value of 38 in register D ensures that the character will not go off the screen. But for horizontal movement, a value of 8 is passed to D to give 64 movements across the screen.

Now then, if you happen to have the very same version of BASIC and you have an 8080A running at 2 MHz, there is no need to worry about any of the software theory . . . just copy the program. Make sure memory size is less than or equal to 16,350. However, if you have a different setup, don't get too involved with the theory presented above. It won't work for you anyhow. I like to be helpful!

At this point you might feel that it would be better to do the whole program in assembly language and not try to marry BASIC to a machine-language routine. As for me, I am satisfied with the program written in BASIC. It is easy to modify if you want to try something else.

Construction Details

In wiring up the joybox, run a 1k resistor from the wiper of one pair of pots to one of the stator lugs. If you find that the pots appear to work backwards (making things go up when they should go down), you don't have to rewire the whole thing. Just move the resistor over to the opposite stator lug. I have put the pots in parallel so that the effective resistance of each pair is 50k Ohms. This helps a little to overcome the irregularities.

An optional SPST push button is used for reset. If you press the push button, the corresponding timer will never reset. Software will wait several milliseconds and then return with a zero value for that channel. The addition of two simple push buttons is almost like adding two more inputs.

To mount the two joysticks, I used a common aluminum box, available in most parts stores. Two four-conductor ribbon cables were taped together to make a cheap eight conductor. Three wires are common, one is +5 volts and the other four run from the timers to their respec-

```
224 DATA CD,38,07,06,70,3E,00,D3
232 DATA 30,7A,3D,C2,EA,3F,00,03
240 DATA 04,CA,FA,3F,DB,70,A3,C2
248 DATA E9,3F,AF,C3,E5,0D,00,00
300 REM START AT HEX 3FE0=63*256+224
310 M=63*256+224
320 FOR N=M TO M+31
330 READ H$:GOSUB 400
340 PRINT N,DD
350 POKE N,DD:NEXT N
360 REM PUT ADR IN USR
370 POKE 73,224
380 POKE 74,63
390 GOTO 490
400 X$=RIGHT$(H$,1)
410 GOSUB 450:DD=X
420 X$=LEFT$(H$,1)
430 GOSUB 450:DD=16*X+DD
440 RETURN
450 X=VAL(X$)
460 IF X$="0" THEN RETURN
470 IF X=0 THEN X=ASC(X$)-55
480 RETURN
490 REM USR PROGRAM FOR JOYSTICK
500 REM FOR ALTAIR 8K BASIC REV 4.0
1000 REM GAME OF STAR CHASE FOR VDM-1
1010 REM REQUIRES JOYSTICKS AND USR ROUTINE
1020 REM REV 1.0  JUN 78  JB PALMER
1030 PRINT:PRINT," S T A R   C H A S E"
1040 PRINT
1050 PRINT"   BETA 107 IS A VERY DENSE AND RARE CLUSTER OF PERFECT"
1060 PRINT"BINARY PAIRS OF MICRO-STARS. THE FEDERATION OF PLANETS"
1070 PRINT"HAS DECLARED THIS AREA OFF LIMITS TO ALL SHIPS."
1080 PRINT"TRAVEL AT SUB-WARP VELOCITY INTO THE GRAVITATION FIELD"
1090 PRINT"OF A BINARY MICRO PAIR WOULD PRECIPITATE ITS DESTRUCTION"
1100 PRINT "    MICRO-STARS CONTAIN LARGE AMOUNTS OF RARE ELEMENTS."
1110 PRINT"STARFLEET HAS LEARNED OF A KLINGON PLAN TO INVADE THE"
1120 PRINT"AREA. THEY WILL CAUSE WIDESPREAD DESTRUCTION."
1130 PRINT"      YOUR MISSION: CAPTURE THE ALIEN VESSEL AND MINIMIZE"
1140 PRINT"DESTRUCTION OF THE MICRO STARS. REMEMBER THAT YOUR"
1150 PRINT"OWN SHIP WILL DESTROY A BINARY MICRO PAIR IF YOU PASS"
1160 PRINT"INTO THE GRAVITATIONAL FIELD AT SUB-WARP VELOCITY."
1170 PRINT"    ONE MORE THING, BETA 107 ALSO CONTAINS BLACK HOLES."
1180 PRINT"THESE WILL APPEAR TO THE SHIP'S SENSORS AS SINGLE"
1190 PRINT"STARS, BUT THEY HAVE A GRAVITATION FORCE THAT WILL"
1200 PRINT"PULL ANY OBJECT INTO THE BLACK ZONE. BE CAREFUL."
1210 PRINT"    STARFLEET OUT."
1220 PRINT
1230 PRINT"SPOCK: CAPTAIN, ABOUT THOSE BLACK HOLES. BETA 107 HAS"
1240 PRINT"ONLY BINARY PAIRS OF MICRO-STARS AND THE BLACK HOLES"
1250 PRINT"WHICH DISPLAY AS SINGLE STARS. LOGICALLY THEN,"
1260 PRINT"ANY STAR WHICH IS NOT A MEMBER OF A BINARY PAIR"
1270 PRINT"MUST BE A BLACK HOLE. UPON ENTERING THE BLACK ZONE"
1280 PRINT"WE WILL LOSE ALL CONTACT. I HAVE PROGRAMMED THE SHIP'S"
1290 PRINT"COMPUTER TO AUTOMATICALLY ENGAGE FULL EMERGENCY POWER."
1300 PRINT"WE WILL THEN BE LOCKED IN ORBIT AROUND THE BLACK HOLE"
1310 PRINT"INSIDE THE BLACK ZONE, THE SO-CALLED AREA OF DARKNESS."
1320 PRINT"TO ESCAPE FROM THE BLACK ZONE IT IS ONLY NECESSARY TO"
1330 PRINT"CHART A COURSE AWAY FROM OUR LAST KNOWN LOCATION"
1340 PRINT"AND JUST WAIT UNTIL WE BREAK FREE."
1350 PRINT"    AND I MIGHT ADD, CAPTAIN, THAT IT WOULD BE FUTILE"
1360 PRINT"TO SPECIFY FULL POWER, INASMUCH AS THE SHIP WILL"
1370 PRINT"ALREADY BE USING FULL POWER TO BREAK FREE. TO DO"
1380 PRINT"SO WOULD CAUSE US TO GO SHOOTING ACROSS THE"
1390 PRINT"GALAXY THE INSTANT WE BREAK FREE.":FOR N=0 TO 999:NEXT N
1400 CLEAR:OUT 200,0:E9=-13312:K9=E9:E0=E9:K0=E9
1410 E1=9729:K4=9732
1420 V=-114:HM=-13426
1430 E2=2050:K8=2056
1440 W=64:E=42:K=35:B=32
1450 X=46:F=0:D=1:G=0:S=58
1460 S9=0:M=-13312:M9=-12288
1470 FOR N=M TO M9 STEP 2:POKE N,B:POKE N+D,B
1480 IF RND(1)<.1 THEN POKE N,S:IF RND(1)<.1 THEN POKE N,X
1490 NEXT
1500 E9=(USR(E1)+V)*W+USR(E2)+HM
1510 IF E0<>E9 THEN POKE E0,B
1520 IF PEEK(E9)=X THEN F=20
1530 IF F THEN F=F-D:GOTO 1560
1540 IF PEEK(E9)=S THEN S9=S9+D
1550 POKE E9,E:E0=E9
1560 K9=(USR(K4)+V)*W+USR(K8)+HM
1570 IF K0<>K9 THEN POKE K0,B
1580 IF PEEK(K9)=X THEN G=20
1590 IF G THEN G=G-D:GOTO 1620
1600 IF PEEK(K9)=S THEN S9=S9+D
1610 POKE K9,K:K0=K9
1620 IF K9<>E9 THEN 1500
1630 FOR N=33 TO 47:FOR N0=0 TO 19:NEXT N0
1640 POKE E9,N:NEXT N:POKE E9,E
1650 PRINT "THE KLINGON HAS BEEN CAPTURED !!"
1660 PRINT "STARFLEET REPORTS "S9" STAR SYSTEMS DESTROYED"
1670 N=INT(S9/7)+1
1680 ON N GOTO 1700,1710,1720,1730
1690 PRINT "YOU REALLY BLEW IT !!":GOTO 1740
1700 PRINT "VERY EXCELLENT WORK, CAPTAIN !"
1710 PRINT "STARFLEET IS SATISFIED WITH YOUR MISSION":GOTO 1740
1720 PRINT "YOU DID A FAIR JOB, BUT-"
1730 PRINT "YOUR NEXT ASSIGNMENT WILL BE ON ALTAIR 2"
1740 INPUT "ANOTHER TRY ";Y$
1750 IF LEFT$(Y$,1)<>"N" THEN 1400
1760 END
2000 PRINT "D<128,E=1,2,4,OR 8"
2010 INPUT"D,E";D,E
2020 A=USR(D*256+E)
2030 INPUT"CHANGE POSITION";R$
2040 B=USR(D*256+E)
2050 PRINT A;B;B-A
2060 GOTO 2010
```

Program B. "Star Chase" listing (for VDM-1 and joysticks).

tive joypots. Shielded cable might reduce some of the interaction between timers, but it seems to work OK as is.

Adjustments

After the minor errors, the joysticks worked with no problem. I have neither fully decoded the I/O port nor made use of the PWR signal on the Altair bus. To date this has not caused any problem. The major weakness is only in the resistance elements of the joysticks. To overcome this problem make the following "adjustment": Tell the players beforehand that there is a strange, unpredictable gravitational distortion in the midst of the Beta 107 star cluster that has been programmed into the computer simulation. This has proved to be very effective. But if you need joysticks for a commercial application, pay quite a bit more and get highly linear elements.

The routine at line 2000 is not part of the game program. It is used to fine-tune the software to correct any variations in hardware. It will input a test value for register D and the mask bit in E.

As an example, put right vertical full up. Enter 38,1 and hit return. Put right vertical full down, then type any letter to the next input statement and hit return. BASIC now gives the first, last and difference for the up and down positions of channel one. For vertical movement on the VDM-1 video display you want just 16 positions, or else the spaceship will go off the screen altogether. Normally the joysticks never return a zero value, so bias is needed. This is the variable V in line 1420.

Variables E1 and K4 are vertical-calling constants: E1 = 256 * D + 1; K4 = 256 * D + 4.

Do the same for the horizontal movements. There are 64 positions across the screen. Now USR must return values that differ by 64 for extreme left or right positions. Again, the minimum value will be the bias. Add the bias value to M to get constant HM in line 1420.

Line 1430 has constants for horizontal movements: E2 = 256 * D + 2; K8 = 256 * D + 8.

Once determined, these values do not have to be changed. Really, they are not at all critical. Improper values will put the spaceships off the screen or give very funny responses. Avoid using BASIC to make calculations within the program. Being an interpreter, it goes rather slowly when doing a lot of calculations.

Using joybox with VDM-1 display.

Two joysticks mounted in aluminum box.

Notice the values for M and M9. This is how the top and bottom of the VDM-1 memory is identified by Mits BASIC. Change these if your VDM-1 is not in the usual memory location. Some other constants are ASCII values for stars or spaceships.

Triumphant Return

Now back to the school with the joysticks! Everybody loved it. My Altair is now a hero; and it also provides a very good exercise in eye-and-hand coordination. You must tell this to the principal if he drops in and asks what your class is doing this period.

Now what the game lacks in intellectual interest is compensated by the excitement it generates. The players are told that a capture is made only by slow movements. Otherwise, a fast movement is a "warp," and the player goes right through an object without disturbing it.

The joystick software can be made more simple or more complex as the user wishes. Most video games today do not allow very slow or very fast movements at the player's option. If that is all you want, forget about the joysticks and just use simple on-off switches. With the joysticks, the player has much more control.

What are some of the future uses of a joystick/video display system? I have a fantasy of a cockpit simulation of the flight of an airplane. What are your ideas?■

References

1. T. R. Buschbach, "An Inexpensive Joystick *Interface*," *Byte*, March 1977.
2. R. Edelson, "Build a 3-Digit A/D Converter for Your Microprocessor," *Interface Age*, January 1977.
3. D. R. Kraul, "Designing Multichannel Analog Interface," *Byte*, June 1977.
4. Don Lancaster, *TTL Cookbook*.
5. Stephen Wozniak, "Build A Simple A to D," *Interface*, November 1976.
6. Altair 8800B documentation —gives info on input and output bus timing.

Attack on the Pack!

Sherm Wantz's handy index/article on page 104 tells you how to "get inside" the TRS-80 manual. Here's an article that tells you what exactly is inside the TRS-80 power pack.

David F. Miller
7462 Lawler Ave.
Niles IL 60648

If you've been wondering (as I had) exactly what is inside the TRS-80 power-pack cube, then this article should be of interest to you! Actually, it was more than just passing curiosity that drove me to brutally tear apart the plastic housing for the power pack (which is the only way I can think of to gain access to the working innards).

My pack, after being on for a while, had (past tense) an extremely distracting transformer lamination buzz that eventually became intolerable. I had heard that other TRS-80 owners had experienced blown fuses (yes, the line fuse is inside the sealed power pack) because many units were produced without Slo-Blo-type fuses, and the standard lag fuses were prone to blow on initial power-up. Knowing my luck, I suspected that my pack most likely did not have a Slo-Blo fuse and that sooner or later I would be tearing (literally) into it. I was right.

The schematic diagram (Fig. 1) of the TRS-80 power pack is basically self-explanatory and should be of some interest to most users. The only additions I made are those shown in the primary circuit (originally it was simply a pigtail fuse and an ac line cord).

If you decide to take the route that I did, check the wiring on your pack to make sure no production changes were made in color coding, etc. Mine has been remounted in an LMB No. C.R.-422 box (4 × 2¼ × 2¼ inches) along with the externally accessible fuse holder and main on-off switch visible in the front view of Photo 1. Photos 2

Photo 1. Front view of the completed power pack.

Photo 2. Inside wiring view of the completed power pack.

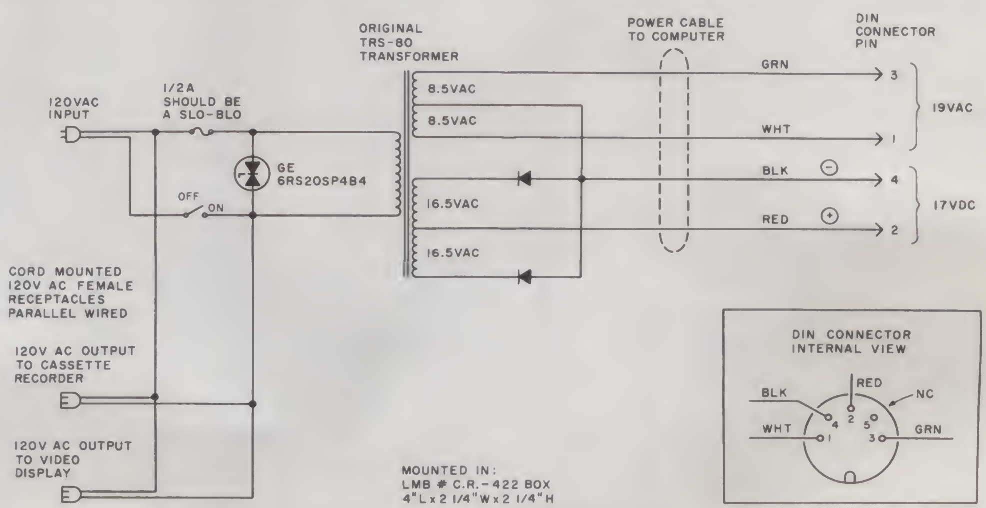


Fig. 1. Modified TRS-80 power pack.

and 3 show inside and end views of the power pack.

The single main switch allows me to turn off the primary power to everything (power pack, video display and cassette recorder), or I can leave the computer itself on and simply shut down the video display with its own power switch if I want to leave a program running but not actually displaying (which I often do). The General Electric 6RS20SP4B4 shown across the transformer primary is a thyrector diode pack that dissipates harmful transients that might come down the ac line from time to time.

My source for this device has been EDI (Electronic Distributors, Inc., 4900 N. Elston Avenue, Chicago IL 60630, (312) 283-4800)—if you are unable to obtain one locally. The thyrector isn't necessary, but I consider it inexpensive insurance against possible transient damage.

The power-pack changes shown in this article are surely no engineering miracle, but they do provide more operating ease and certainly more reasonable access to the individual components of the pack. If you take your time in dissecting the original pack, you should have little difficulty in finishing this project in one evening. ■

Photo 3. End view of the completed power-pack wiring. Note the transformer at the very rear of the photo.

✓ Reader Service—see page 179

NORTH STAR

DISK ASSEMBLER and DISK EDITOR

Both programs read and write disk files; file size not limited by memory. Assembler will assemble up to ten source files at a time; permits modular programming with programs easily relocated by reassembling at the desired address. Editor does not use line numbers; it searches for strings. Lines may be inserted, deleted and displayed. Large disk source files allow programs to be fully commented.

ASSEMBLER/EDITOR on disk with users manual...$30

COMPUTER SYSTEMS DESIGN
906 N. Main Wichita, KS 67214

DEALERS INQUIRIES INVITED

LEVEL IV

32020 Cheboygan ✓L19
Westland, Mich. 48185 **(313) 326-0386**

FOR THE PROFESSIONAL USE OF THE TRS-80™*

BUSINESS PROGRAMS

General Ledger
Single Entry type, permits you to keep track of your incoming and outgoing checks.......... **$99.00**

Mailing Lister
Permits you to keep your clients all in one file and select up to 5 different types of mailings to send out. Both lables and clients journal. 2 types of sorts, zip and name.......... **$49.00**

Inventory Control
Giving you printouts of in-stock, and cost breakdown of orders. Also keeps track of vendors, min. orders and more.......... **$99.00**

All of the above programs are on disk and come with full instructions. All programs are user orientated, so you don't have to be a programmer.

GAMES

Micro Chess
By Peter Jennings. The best computer game for the TRS-80™.......... **$19.95**

Space Battle
Based on Star Trek but better. This may be the most-wanted game for the TRS-80™. A graphic war game in Level II. This copyrighted game is available in Level I. This is the original one written by Rick Papo.......... **$10.00**

Final Approach
Aircraft simulator. In full graphics brings this jet plane down to the ground, but watch out for the small problems that come up, like a cabin fire and more. Level I and Level II on one tape.......... **$10.00**

Galaxy Target
Shoot down the alien machine but don't lose all your power. Bonus game on the back side, both Level II 4K.......... **$10.00**

UTILITIES

Renumber
Renumber your Level II programs. Gives you a printout to the screen or line printer, also.......... **$20.00**

ASCII Compare
Compares two different ASCII stored program files. Producing a third file of the differences. Great when version two or your program doesn't work and you forgot what you changed.......... **$5.00**

Newlist
Display ASCII Files to screen or printer (60 lines per page). If file is basic program, it will break each line to individual instructions.......... **$5.00**

Where Was It
Examines an ASCII file (program) for key words you select (up to 100 different). Useful for conversions from one basic to another.......... **$5.00**

Level I Relocated In Level II RAM
Have the best of both worlds. A Level I machine and a Level II in the same machine without avoiding your warranty. Cassette or disk.......... **$20.00**

All the above Utilities programs are sold on disk (disk $5.00 extra)

Hardware

Memory for TRS-80™ 4116's Full Instructions and Dip Shunts.......... $109.00
TRS-80™ Schematic for Keyboard.......... $5.00
TRS-80™ Interface Schematic.......... $5.00
Diskette . . . Each.......... $5.00
Diskette . . . Box of 10.......... $45.00
TRS-80™ Keyboard and Monitor/Interface Cover.......... $15.00

Please add 75¢ Postage per Order. Michigan Residents add 4% Sales Tax.
FOR MORE INFORMATION ON ANY PROGRAM JUST WRITE

NOW A SOLUTION TO YOUR I/O HEADACHE #33

The OE 1000 Terminal is a low cost stand alone video terminal that operates quietly and maintenance free. It will allow you to display on a monitor or modified T.V. 16 lines of 64 characters. The characters can be any of the 96 ASCII alphanumerics, and any of the 32 special characters. In addition to upper-lower case capability it has a scroll up feature and full X-Y cursor control. All that is required from your microcomputer is 300 baud, RS 232 or 20 mA current loop, serial data. And if that is not enough the price is only $275.00 in kit or $350.00 assembled, plus $5.00 shipping and handling. To order phone or write:

OTTO ELECTRONICS ✓09
P.O. Box 3066
Princeton, N.J. 08540
609/448-9165

MC, BAC, COD accepted N.J. residents add 5% sales tax.
Dealer Inquiries invited.

TRS - 80 OWNERS

When you get tired of backgammon and space treck you'll want to put your computer to more serious uses. Perhaps we can help.

Our business is the generation of well-written, well-documented scientific software. Our first offering is a reliable, accurate general purpose matrix inversion program which can be used as a stand alone mainline or can be folded into your own program (source languages are level I and level II basic). The level I routine is tightly packed and both allow the inverse to be built up over the original matrix to conserve memory if you desire. A 10x10 runs about 45 sec. A 16K machine can handle a 60x60 array. The distribution medium is cassette. The price is $9.95 prepaid with order. (N.Y. residents add 7% sales tax.)

Other programs are now in preparation. Watch our ads or drop us a line if you have a specific requirement.

SIGNIFICANT SOFTWARE
✓S73
Box 9274
ROCHESTER, N.Y. 14625

✓ Reader Service—see page 179

Reader Service—see page 179

8K BASIC for KIM-1

The famous high speed *Microsoft* BASIC is now available for KIM-1. The 9-digit accuracy version is available on cassette or paper tape.

DELIVERY FROM STOCK.

$99.00

$2.00 shipping and handling

JOHNSON COMPUTER J4

P.O. BOX 523
MEDINA, OHIO 44256 (216) 725-4560

the ULTIMATE in CHEAP VIDEO

BOOK & KIT ONLY $42.95

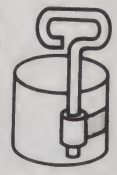
Don Lancaster's "Cheap Video" concept allows almost unlimited options, including:

* Scrolling · Full performance cursor.
* Line/Character formats of 16/32, 24/80, 32/64.... or almost anything.
* Graphics - up to 256 X 256 B&W; 96 X 128 COLOR (requires low-cost option modules)
* Works with 6502, 6800 and other micros.

SPECIAL OFFER: Buy the Kit (upper case alpha-numeric option included) & get the Book at 1/2 price.

P9 PAiA ELECTRONICS, DEPT. 2-K, 1020 W. WILSHIRE BLVD., OKLAHOMA CITY, OK 73116

I'm Sold, PLEASE RUSH..... ( ) SEND FREE CATALOG
( ) TVT-6 5/8 Kit & Cheap Video Cookbook - $42.95 (enclosed)
( ) TVT-6 5/8 Kit only (book required for assembly) - $39.95
name: ______
address: ______
city: ______ state: ______ zip: ______
PAiA ELECTRONICS DEPT. 2-K, 1020 W. WILSHIRE BLVD., OKLAHOMA CITY, OK 73116



canned software

TRS-80

SOFTWARE!

YES! QUALITY SOFTWARE FOR THE TRS-80 IS NOW WRITTEN & AVAILABLE. BCC IS PLEASED TO BE ABLE TO PRESENT SOME VERY FINE SOFTWARE NOW WITH EVEN MORE AVAILABLE IN THE VERY NEAR FUTURE. ALSO WE DEVELOP CUSTOM DESIGNED SOFTWARE FOR YOUR EVERY NEED. WRITE US FOR A FREE PRICE QUOTE.

FOR SOFTWARE THINK BCC

O EXPENSE REPORTING SYSTEM
LEVEL II 16K $22.95
Keeps track of up to 50 Expense Categories for a full year. Gives monthly, quarterly, and annual totals of expenses. For the small business or home use.

O BACKGAMMON
LEVEL I or II 16K $19.95
PLAY AGAINST THE COMPUTER & LOSE! The computer plays a darned good game. With Level II you can even bet!

O SPACETREK-80
LEVEL I or II 16K $9.95
Play a strange game of blowing-up stars to find starbases. Watch out for those antimatter mines! Tune-in on your AM radio for strange sound effects.

FOR SUPPLIES THINK BCC

O C20 CASSETTES
The same high-density, calendered tape we use for our own programs.
1-12 $1.59ea. 13-24 $1.29ea.
25+ $1.19ea.

O CODING PADS
BASIC LANGUAGE CODING PADS $.70
50 sheets of 20 lines per sheet and 64 characters per line.
SCREEN FORMAT PADS $1.00
50 sheets per pad

ORDER NOW! All programs come on quality cassettes. Programs are available on diskette for an additional charge of $7.00 per order. *NYS residents add 7%.*

Check boxes of items being ordered:
# of items ordered______ Total amount enclosed______(All items must be prepaid.)

Bourrut Consulting Corporation

B35

**21 friendly rd.
smithtown ny 11787**

Dept. TRS - 80

Reader Service—see page 179

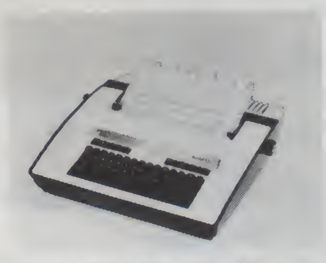
THE END OF YOUR SYSTEM

Put this SOROC IQ 120 at the end of your system for fast and silent input/output.

795^{00*}

Put this T.I. 810 printer at the end of your system for fast and reliable hard-copy output.

1695^{00*}



Put this TELETYPE 43 at the end of your system for reliable low-cost input/output.

999^{00*}

*plus 2% handling. Shipped freight collect. ✓M73

MICROMAIL™

MICROMAIL • BOX 3297 • SANTA ANA, CA 92703
(714) 731-4338

Also DIABLO, LEAR/SIEGLER, DEC
Write or call for a free catalogue

HOBBYISTS! ENGINEERS! TECHNICIANS! STUDENTS!

Write and run machine language programs at home, display video graphics on your TV set and design microprocessor circuits—the very first night—even if you've never used a computer before!

ELF II featuring RCA COSMAC microprocessor/mini-COMPUTER $99^{95}

Stop reading about computers and get your hands on one! With ELF II and our new *Short Course* by Tom Pittman, you can master computers in no time at all! ELF II demonstrates all 91 commands an RCA 1802 can execute and the *Short Course* quickly teaches you how to use each of the 1802's capabilities.

ELF II's video output lets you display an alphanumeric readout or graphics on any TV screen or video monitor plus enjoy the latest video games, including an exciting new target/missile gun game that was specifically developed for ELF II.

But that's not all. Once you've mastered computer fundamentals, ELF II can give you POWER with add-ons that are among the most advanced found anywhere. No wonder IEEE chapters plus hundreds of universities and major corporations have chosen the ELF II to introduce their students and personnel to microprocessor computing!

Learn The Skill That May Soon Be Far More Important Than Your College Degree!

The ability to use a computer may soon be more important to your earning power than a college degree. Without a knowledge of computers, you are always at the mercy of others when it comes to solving highly complex business, engineering, industrial and scientific problems. People who understand computers can command MONEY and to get in on the action, you must learn computers. Otherwise you'll be left behind.

ELF II Is The F-A-S-T Way To Learn Computer Fundamentals!

Regardless of how minimal your computer background is now, you can learn to program a computer in almost no time at all. That's because Netronics has developed a special *Short Course on Microprocessor And Computer Programming* in non-technical language that leads you through every one of the RCA COSMAC 1802's capabilities so you'll understand *everything* ELF II can do . . . *and how to get ELF II to do it!*

All 91 commands that an 1802 can execute are explained to you, step-by-step. The text, written for Netronics by Tom Pittman, is a tremendous advance over every other programming book in print.

Keyed specifically to the ELF II, it's loaded with "hands on" illustrations. When you're finished, ELF II and the 1802 will no longer hold any mysteries to you.

In fact, not only will you be able to use a personal computer creatively, you'll also be able to read magazines such as BYTE . . . INTERFACE AGE . . . POPULAR ELECTRONICS and PERSONAL COMPUTING and *understand* the articles.

If you work with large computers, ELF II and our *short Course* will help you to understand what makes them tick.

A Dynamite Package For Just $99.95!

With ELF II, you learn to use machine language—the fundamental language of all computers. Higher level languages such as FORTRAN and BASIC must be translated into machine language before a computer can understand them. With ELF II you build a solid foundation in computers so you'll really know what you're doing, no matter how complicated things get.

Video output also makes ELF II unique among computers selling for such a low price. Attached to your TV set, ELF II becomes a fabulous home entertainment center. It's capable of providing endless hours of fun for both adults and children of all ages! ELF II can create graphics, alphanumeric displays and fantastic video games.

No additional hardware is required to connect ELF II to your TV's video input. If you prefer to connect ELF II to your antenna terminals instead, simply use a low cost RF modulator (to order one, see coupon below).

ELF II's 5-card expansion bus (connectors not included) allows you to expand ELF II as your needs for power grows. If you're an engineer or hobbiest, you can also use ELF II as a counter, alarm, lock, thermostat, timer or telephone dialer, or for countless other applications.

ELF II Explodes Into A Giant!

Thanks to ongoing work by RCA and Netronics, ELF II add-ons are among the most advanced anywhere. Plug in the **GIANT BOARD™** and you can record and play back programs, edit and debug programs, communicate with remote devices and make things happen in the outside world. Add **Kluge Board** to get ELF II to solve special problems such as operating a more complex alarm system or controlling a printing press. Add **4k RAM** board and you can write longer programs, store more information and solve more sophisticated problems.

Expanded, ELF II is perfect for engineering, business, industrial, scientific and personal finance applications. No other small computer anywhere near ELF II's low price is backed by such an extensive research and development program.

The **ELF-BUG™ Monitor** is an extremely recent breakthrough that lets you debug programs with lightening speed because the key to debugging is to know what's inside the registers of the microprocessor and, instead of single stepping through your program, the ELF-BUG™ Monitor, utilizing break points, lets you display the entire contents of the registers on your TV screen at any point in your program. You find out immediately what's going on and can make any necessary changes. Programming is further simplified by displaying 24 bytes of RAM with full address, blinking cursor and auto scrolling. A must for serious programmers!

Netronics will soon be introducing the ELF II **Color Graphics & Music System**—more breakthroughs that ELF II owners will be the first to enjoy!

Now BASIC Makes Programming ELF II Even Easier!

Like all computers, ELF II understands only "machine language"—the language computers use to talk to each other. But, to make life easier for you, we've developed an **ELF II Tiny BASIC.** It talks to ELF II in machine language for you so that you can program ELF II with simple words that can be typed out on a keyboard such as PRINT, RUN and LOAD.

"Ask Now What Your Computer Can Do . . . But *What Can It Do For YOU!*"

Don't be trapped into buying a dinosaur simply because you can afford it and it's big. ELF II is more useful and more fun than "big name" computers that cost a lot more money.

With ELF II, you learn to write and run your own programs. You're never reduced to being a mere keypunch operator, working blindly with someone else's predeveloped software.

No matter what your specialty is, owning a computer which you really know how to use is sure to make you a leader. ELF II is the fastest way there is to get into computers. Order from the coupon below!

SEND TODAY!

NOW AVAILABLE FOR ELF II—

☐ Tom Pittman's ***Short Course On Microprocessor & Computer Programming*** teaches you just about everything there is to know about ELF II or any RCA 1802 computer. Written in non-technical language, it's a learning breakthrough for engineers and laymen alike. **$5.00** postpaid!

☐ Deluxe metal **cabinet** with plexiglas dust cover for ELF II, **$29.95** plus $2.50 p&h.

☐ ELF II connects to the video input of your TV set. If you prefer to use your antenna terminals, order **RF Modulator, $8.95** postpaid.

☐ **GIANT BOARD™** kit with cassette I/O, RS 232-C/TTY I/O, 8-bit P I/O, decoders for 14 separate I/O instructions and a system monitor/editor, **$39.95** plus $2 p&h.

☐ **Kluge** (Prototype) **Board** accepts up to 36 IC's. **$17.00** plus $1 p&h.

☐ **4k Static RAM** kit. Addressable to any 4k page to 64k. **$89.95** plus $3 p&h.

☐ Gold plated **86-pin connectors** (one required for each plug-in board). **$5.70** postpaid.

☐ Professional **ASCII Keyboard** kit with 128 ASCII upper/lower case set, 96 printable characters, onboard regulator, parity, logic selection and choice of 4 handshaking signals to mate with almost any computer. **$64.95** plus $2 p&h.

☐ Deluxe metal **cabinet** for ASCII Keyboard, **$19.95** plus $2.50 p&h.

☐ **ELF II Tiny BASIC** on cassette tape. Commands include SAVE, LOAD, ±, ×, ÷, ( ), 26 variables A-Z, LET, IF/THEN, INPUT, PRINT, GO TO, GO SUB, RETURN, END, REM, CLEAR, LIST, RUN, PLOT, PEEK, POKE. Comes fully documented and includes alphanumeric generator required to display alphanumeric characters directly on your TV screen without additional hardware. Also plays tick-tack-toe plus a drawing game that uses ELF II's hex heyboard as a joystick. 4k memory required. **$14.95** postpaid.

☐ Tom Pittman's **Short Course on Tiny BASIC** for ELF II, **$5** postpaid.

☐ **Expansion Power Supply** (required when adding 4k RAM). **$34.95** plus $2 p&h.

☐ **ELF-BUG™ Deluxe System Monitor** on cassette tape. Allows displaying the contents of all registers on your TV at any point in your program. Also displays 24 bytes of memory with full addresses, blinking cursor and auto scrolling. A must for the serious programmer! **$14.95** postpaid.

Coming Soon: A-D, D-A Converter, Light Pen, Controller Board, Color Graphics & Music System . . . *and more!*

Call or write for wired prices!

Netronics R&D Ltd., KB ✓N15
333 Litchfield Road, Phone
New Milford, CT 06776 (203) 354-9375

Yes! I want to run programs at home and have enclosed: ☐ **$99.95** plus $3 postage & handling for **RCA COSMAC ELF II kit,** ☐ $4.95 for power supply (required), ☐ $5 for RCA 1802 User's Manual, ☐ $5 for *Short Course on Microprocessor & Computer Programming.*

☐ **I want mine wired and tested with power supply, RCA 1802 User's Manual and *Short Course* included for just $149.95 plus $3 p&h!**

☐ I am also enclosing payment (including postage & handling) for the items checked at the left.

Total Enclosed (Conn. res. add tax) $______ ☐ Check here if you are enclosing Money Order or Cashier's Check to expedite shipment.

USE YOUR ☐ VISA ☐ Master Charge
(Interbank #____ ____ ____ ____)
Account # ________
Signature ________ Exp. Date ______

PHONE ORDERS ACCEPTED (203) 354-9375

Print Name ________
Address ________
City ________
State ______ Zip ______

DEALER INQUIRIES INVITED

✓ Reader Service—see page 179

Reader Service—see page 179

✓ Reader Service—see page 179

DO THESE NUMBERS MEAN ANYTHING TO YOU?

44 PINS

4.5" x 6.5" CARDS

5 VOLTS

Wasatch delivers quality system components from stock

	Assem./Tested	Kit	PCB
• 4K x 8 RAM	$89.95	$79.95	$22.50
• 2K/4K x 8 PROM		79.95/129.95	22.50
• 8-SLOT MOTHER	50.00	40.00	20.00
• 80 LINE DIGITAL I/O		59.95	20.00

New 5V system components under development

- 8K x 8 EPROM (2716)
- 16/32 Channel, 8 Bit A-D
- 4 Channel, 8 Bit D-A

Wasatch also delivers prime semiconductor memory at reasonable price, in addition to custom design services.

4K RAM

W13

SEND CHECK OR MONEY ORDER

WASATCH SEMICONDUCTOR PRODUCTS

25 South 300 East • Suite 215 • Salt Lake City, Utah 84111

For orders under $25 add $2 shipping & handling / Utah residents add 5 sales tax

STARFLEET ORION

A specially designed SF TACTICAL BATTLE GAME for your PET or TRS-80 Computer.
Take command of a star fleet as you seek to drive your opponent's spaceships from the skies. Rescue a crippled battlecruiser near the planet Mardak. Lead a swarm of swift fighters against an almost invulnerable enemy battle station. Create your own battle scenarios based on your favorite SF stories or from your own imagination. Missiles and torpedos explode all about your ships while great energy beam projectors lock on target and batter against the ship's shields. Will you or the enemy prevail!

STARFLEET ORION is a complete game system

- rule book
- battle manual
- casette
- ship control sheets
- program listings

Includes 2 programs, 22 space ship types, and 12 playtested scenarios. Game mechanics are extremely simple, but play is exciting, challenging, and rich in detail. **When ordering, specify Commodore PET (8K) or TRS-80 (level II) version.**

Send check or money order for $16.95 to:

Automated Simulations A71
Department B
P.O. Box 4232
Mountain View, CA. 94040

California residents please add 6% sales tax

INNOTRONICS

IF QUALITY IS FOREMOST IN YOUR REQUIREMENTS WE ARE PROUD TO OFFER THE HIGHEST QUALITY DISKETTE DRIVE FOR THE FAIREST PRICE

NEW MODEL 410, 8 INCH FLOPPY DISK DRIVES **$495**
TWO DRIVE SUB-SYSTEMS (less controller) **$1555**

INNOTRONICS CORPORATION DEPT D I22
BROOKS ROAD
LINCOLN, MASSACHUSETTS 01773
Tel. (617) 259 0600

Radio Shack Computer Users

TRS-80 monthly newsletter

The largest publication devoted to the TRS-80 System

- **Business**
- **Personal Finance**
- **Practical Applications**
- **Gambling—Games**
- **Latest RADIO SHACK Developments**
- **Software Exchange**
- **Market Place**
- **Questions and Answers**
- **Program Printouts**
- **. . . and more**

Major programs published monthly . . . Complete income tax program (long and short forms) . . . Inventory control . . . Extensive mailing list and file program . . . Payroll . . . Stock selection and indicators . . . Horse selector for picking winners . . . Renumber program lines . . . Chess . . . Checkers . . . Financial package . . .

$24. Per Year

VISA master charge

MATHEMATICAL APPLICATIONS SERVICE M64

Box 149K New City, New York 10956 (914) 425-1535

Send for FREE Software Catalogue (Including listings of hundreds of TRS programs available on cassette and diskette).

KIMSI

The KIM to S-100 bus Interface/Motherboard

- Combines the power of the 6502 with the flexibility of the S-100 bus
- Attaches to any unmodified KIM
- Complete interface logic and fully buffered motherboard in one unit
- On-board regulation of power for KIM
- Eight slots of S-100 compatibility for additional RAM, Video and I/O boards, PROM Programmers, Speech processors . . .
- Includes all parts, sockets for ICs, one 100 pin connector, and full Assembly/Operating documentation

◆ *Kit $125, Assembled $165*
◆ *All units shipped from stock*

FORETHOUGHT PRODUCTS
87070 Dukhobar Rd. #H
Eugene, OR 97402 (503) 485-8575 ✓F8

You're gonna LOVE this!

16-K Apple II
with SUP 'R' MOD II & 1 Cassette
OUR USUAL GREAT LOW PRICE! **$1195. ea.**

LEAR-SIGLER 80M - 3A Assembled
REG. $895. NOW! **$825. ea.**

KIM-1 REG. $179.95 **$159.95 ea.**

KIM-1 Power Supply
REG. $39.95 NOW! **$29.95 ea.**

HAZELTINE 1500
REG. $1225. NOW! **$1125. ea.**

HAZELTINE 1400
REG. $850. NOW! **$775. ea.**

DEMO EXCELLENT COND. ONLY ONE!
POLY 8813 with 32-K and PRINT INTERFACE
REG. $4985 NOW! **$3985.**

POLY SYSTEM 6 (BRAND NEW!)
REG. $1575. BELOW COST! **$1100. ea.** ORDER NOW ONLY 4 IN STOCK

SANYO VM-4209 9 Inch Monitor
REG. $240. NOW! **$175.**

VERBATIM 5¼" DISKETTES IN PLASTIC CASE
NOW! **$3.95 ea.** MINIMUM ORDER $10.

✓C64

APPLE II or TRS-80 16-K MEMORY CHIPS LOW, LOW PRICE! **8 for $99.**

MAIL or TELEPHONE ORDERS ONLY!
No C.O.D.'s

Master Charge, Visa or Check.
AVAILABILITY: STOCK TO 30 DAYS
OFFER ENDS MARCH 14, 1979

Computer Corner
OF NEW JERSEY
240 WANAQUE AVE.
POMPTON LAKES, N.J. 07442
(201) 835-7080

30– TRS-80—TRS-80—TRS-80—TRS-80—TRS-80—TRS-80—TRS-80—TRS-80 –T

software from ACS service

1. *Z-80 DISASSEMBLER:* *$20.00*

Shows the symbolic code for the machine instructions stored in the memory of your TRS-80. Displays addresses and machine code in hexadecimal, ASCII representation, and symbolic instructions, with operands, on video monitor or line printer. Decodes all Z-80 instructions! Zilog mnemonics used. Code can be reassembled using the TRS-80 Editor/Assembler.

REQUIRES:
Level I or II. We have a version for all versions of the TRS-80 (please state which one you own).

2. *DATA BASE MANAGEMENT:* *$39.00*

This is a complete Data Base Management Program for the TRS-80 Disk System. It employs five commands: Find, Add, Change, Video and Print. You can name your own headings for all fields and can store any type of information for quick retrieval. All headings and data are kept on disk. Easy to use but professional. Example of use: Store index of magazine articles so you don't have to flip through all of your computer magazines to find an article.

REQUIRES:
Level II Disk Basic and one disk drive, 16K RAM, comes on cassette, for disk add $7.50.

3. *10 KEY NUMERICAL KEY PAD.*

Ribbon Cable plugs into keyboard or expansion interface. No modifications necessary.

4. *COMPREHENSIVE MEMORY TEST:* *$7.95*

Routines for all Level II TRS-80s.

5. *INVENTORY:* *$20.00*

Uses sequential files on disk to store inventory. You can list stock number, item name, location, how many, cost per unit, number per case, cost per case, and next shipping date. Commands include: Check for item, change item info, add new items, and print entire inventory to line printer. Can be used without line printer. Easy to use, just load and run, type in your inventory and you are ready for quick retrieval of any item.

REQUIRES:
Level II disk drive and basic, 16K RAM.

6. *LEVEL III BASIC ON CASSETTE* *$30.00*
7. *T-BUG ON DISK* *$20.00*

NOTE:
All programs came on cassette unless noted. If you want it on disk, please specify so and add $7.50 to your order, or send a diskette with your order. All orders shipped same day. All programs guaranteed to run.

MASTER CHARGE & VISA WELCOME

AUTOMATED COMPUTER SOFTWARE SERVICE ✓A75
2208 DEARBORN DR.
DONELSON, TN 37214
615-883-4800

---

22 START-AT-HOME COMPUTER BUSINESSES

In "The Datasearch Guide to Low Capital, Startup Computer Businesses"

CONSULTING • PROGRAMMING • SOFTWARE PACKAGES • COM • FREELANCE WRITING • SEMINARS • TAPE/DISC CLEANING • FIELD SERVICE • SYSTEMS HOUSES • LEASING • SUPPLIES • PUBLISHING • TIME BROKERS • HARDWARE DISTRIBUTORS • SALES AGENCIES • HEADHUNTING • TEMPORARY SERVICES • USED COMPUTERS • FINDER'S FEES • SCRAP COMPONENTS • COMPUTER PRODUCTS AND SERVICES FOR THE HOME.

Plus -- Loads of ideas on moonlighting, going full-time, image building, revenue building, bidding, contracts, marketing, professionalism, and more. No career planning tool like it. Order now. If not completely satisfied, return within 30 days for full immediate refund.

• 8½ x 11 ringbound • 156 pp. • $20.00

Phone Orders **901-382-0172**

DATASEARCH ✓D40
incorporated
5694 Shelby Oaks Dr., Suite 105, Dept. A, Memphis, TN 38134

Rush____copies of "Low Capital Startup Computer Businesses" at $20 per copy to me right away.

NAME/COMPANY __________
ADDRESS __________
CITY/STATE/ZIP __________

☐ Check Enclosed ☐ Bankamericard ☐ Master Charge

\# __________

NORTH STAR USERS

Develop your own custom files for data storage and retrieval without programming. Print your own customized reports from the data without programming with FLOWDATA's Universal General File System. Any type of data from Addresses to Zoology. Files and reports limited only by your imagination. NO PROGRAMMING SKILL REQUIRED. $50.00 on North Star Disk.

Also Available—Turtle Driver—Northshare Enhancements (2 keyboards, 2 CRT)—General Ledger—Accounts Receivable and Payable—Modem Drivers—Special Operating Systems—Games (Bowling, Kalah, Crypto, ICBM and others)

Write:

FLOWDATA

4503 Seaway Drive
Lansing MI 48910 ✓F11

TRS-80 & P.E.T. SOFTWARE
AVAILABLE NOW: FUNCTIONAL SOFTWARE
for the Entrepreneur, Salesman or Market Trader

Three separate packages @$19.95/each include cassette and hard copy documentation.

1. ACCOUNTING SYSTEM—Includes 2 programs: Program #1 for ledger postings by account with automatic 2nd entry made by the computer. Program #2 for analysis of data file (balance sheet + income/expense statement)
2. STOCK MARKET ANALYSIS SYSTEM—Includes 2 programs: Program #1 for establishing historical data file. Utilizing technical market analysis techniques. Program #2 analizes historical data and current data issuing buy and sell signals.
3. STOCK MARKET FINANCIAL SYSTEM—Includes 2 programs: Program #1 for data file control. Program #2 offering 8 analysis option, looks at your stock and option transactions.

Distributed by: STEVE SHAW P.E. ✓S76
Software Consultant
P.O. Box 1707
Tampa, Florida 33601

UNEXPLAINED GLITCHES/CRASHES?

AC Power Line surges and hash could be the culprit! Our Surge Suppressors and Hash Filters curb damaging surges and disruptive hash. Send self-addressed, stamped envelope for FREE Surge and Hash Cures flyer. Suppressor/Filter at your Dealer or direct.

S/F-KW-3 1000 watt line cord unit. $22.95
ELF/T-10 10 amp wire-in unit. $18.50

NEW ! ! ISOLATOR. $49.95

3 filtered 3 pr. sockets reduce printer/processor interaction 1 KW each socket; 1875 watt total load handling.

Phone Orders Welcome
1-617-655-1532

VISA master charge

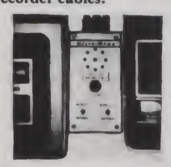
ELECTRONIC SPECIALISTS

DEPT. KB ✓E36
171 S. Main St., Natick MA 01760

TRS-80

Micro-Mega CASSETTE CONTROL UNIT

● Speed up your cassette tape handling ● Pinpoint program locations on tape with an audible auxiliary monitor ● Get protection from recording and playback glitches resulting from ground loops ● Eliminate the tedious plugging and unplugging of recorder cables.



The Micro-Mega Cassette Control Unit does all this and more. You get instant manual control of the recorder at the flick of a switch. Want to find the beginning or end of a program? Flick another switch, and you'll hear it. All cables remain plugged in all the time.

The Micro-Mega Cassette Control Unit does a lot to clean up and neaten up your whole TRS-80 installation, too. As shown at the left, it's in a 2½" x 5" box which snuggles between the keyboard and your recorder. There is no need to move the recorder, and there no longer are any 90 degree cable bends whipping around. It's fast, neat and convenient—and it's a bargain!

CASSETTE CONTROL UNIT—$32.50

Micro-Mega ✓M54

P.O. BOX 6265 ● ARLINGTON VA 22206
(Virginia residents add 4% sales tax)

ANNOUNCING ¿COMPUPRISM?

CHECK THIS LITTLE MUTANT OUT ! ! !

Single board color graphics interface for s-100 bus systems.
16K on board memory ! !
16 colors always available !
27648 individual picture elements !
Every element programmable in any color at any time !
Sequential memory mapping makes programming easy !
Comes assembled and tested !

ONLY $250!

GET THE MOST FOR YOUR MONEY ✓J13

J.E.S. GRAPHICS
P.O. BOX 2752
TULSA, OK 74101
[918]-742-7104

INTECOLOR COMPUCOLOR

SALES ● SOFTWARE
SERVICE ● UPDATES
—BUY—SELL—SWAP—

WHOLESALE—to dealers
UPDATE—your machine to the latest disk configuration
SERVICE—providing local and west coast service for east coast firms
SOFTWARE—over 500 programs in our library
USERS GROUP—join to purchase low cost software
MEMBERSHIP & BULLETINS—1 year—$25
PROGRAM EXCHANGE—5 to 10 programs for each acceptable submission

S. P. ELECTRONIC SYSTEMS
5250 Van Nuys Blvd.
Van Nuys CA 91401 ✓S75
Phone 213/788-8850

COMPUTER FORUM ✓C111

14052 E. Firestone Blvd.
Santa Fe Springs CA 90670 (213) 921-2111 (714) 739-0711

● BUSINESS ● EDUCATIONAL ● PERSONAL

—H—A—R—D—W—A—R—E—

★ APPLE ★ NORTHSTAR
★ CENTRONICS ★ HEURISTICS
★ INTROL ★ VIM

—S—O—F—T—W—A—R—E—

★ APPLE II ★ NORTHSTAR
★ PET ★ TRS-80

—SERVICE DEPT.—CLASSES—

—APPLE BYTE USERS GROUP—
Meets on 2nd Saturday each month
—PROGRAMMERS GROUP—
Meets on 1st Saturday each month

★ Send for our hardware catalog, flyers and software list.

SLAVEMASTER

2650 μP MULTIPROCESSOR

Realtime control or development system. Slave & Master Microprocessors share S100 Memory & I/O. Uses two identical cards, each with the following features:

- Signetics 2650 μP
- KC Cassette Interface
- 20 MA/EIA Port 110/300/1200
- 8 Vectored Ints
- Keyboard Int
- Realtime Int
- Power Fail Int
- Power on Reset/Jump
- 4 2708 EROM Sockets
- Shared Memory or Split, with Mailbox

Software: Signetics "PIPBUG", Signetics "LITTLE ASSEMBLER", and REAL TIME CTL.

$269.00 assembled, $198.00 kit.
Send for specifications.

VICTORIA MICRO DIGITAL
401 Dundee Street ✓V17
Victoria TX 77901

S-100 11 Slot & others

Card Cage

For 8803 Motherboard

$34.50

● Post Paid USA ●

Complete with Edge Guides And Hardware
Dealer Discount Club Refund

M.W. Keith & Co.

P.O. Box 954 ✓K15
● Racine Wisconsin 53405 ●

Kilobaud Classified

Kilobaud Classified advertisements are intended for use by persons desiring to buy, sell or trade used computer equipment or software. No commercial ads are accepted.

Two sizes of ads are available. The $5 box allows up to 5 lines of about 37 characters per line, including spaces and punctuation. The $10 box allows up to 10 lines. Minimize use of capital letters to save space. No special layouts allowed. Payment is required in advance with ad copy. We cannot bill or accept credit.

Advertising text and payment must reach us 60 days in advance of publication (i.e., copy for March issue, mailed in February, must be here by Jan. 1). The publisher reserves the right to refuse questionable or not applicable advertisements. Mail copy with payment to: Kilobaud Classified, Kilobaud, Peterborough NH 03458. Do not include any other material with your ad as it may be delayed.

TRS-80/POLY/PET software: Bomber, Biortm, Lander use graphics, fits 4K, on tape—$9.90. Request FREE catalog. (Specify computer type.) G. F., Jester Cntr-W58, Austin TX 78784.

We had replacement panels made for our Altair 8800s. They're blue against a white background and really look nice! We'll sell the extras for $19 each. D.L. Tibbetts, 520 Schoolhouse Ln., Willow Grove PA 19090.

TRS-80 Software: Business programs written in 16K Level II. Available for cassette or disk files. Send for list & prices. L. Owens, Rt. 6, Box 336A, Thomasville GA 31792.

Educational software and other useful programs on cassette for TRS-80, Level II. Descriptions from Dr. S. Harter, Box 17222, Tampa FL 33682.

Sophisticated 2-player football game on cassette for TRS-80 Level II. Uses graphics to draw field, move ball, etc. Dr. S. Harter, Box 17222, Tampa FL 33682.

TRS-80 Owners! Tired of endlessly thumbing through your Level II manual? Send $1 and SASE and receive a complete (and I *do* mean complete) TRS-80 Level II Manual Index. You won't believe how much time this will save you. *Everything* is listed alphabetically. Send $1 to Dave Rose, 196 Governors Drive, Forest Park GA 30050.

Heath H-8 system. 16K RAM; 2 cassettes; parallel I/O; H-9 video display, all systems tapes and many program tapes. Fully assembled & working, $900. Also, Intertec Data INTERTUBE video terminal RS-232, 24 × 80 upper/lowercase & graphics. New, $650. I will trade all for a good RS-232 IBM wide-platen Selectric IO writer. Steve Vicker, PO Box 1548, Panama City FL 32401. (904) 769-4492 M-Sat.

PET-2001-8 Music kit: Install the amp and speaker in two hours. Drill only one hole above fuse. Software and all hardware inc. $14 & $1 postage. Lou Ensten, Box 6, Twinsburg OH 44087.

TRS-80 software on tape. Games and more with documentation. Also some hardware offerings. Send SASE for more info. Liberty Ridge Homestead, RR 1, Box 44B, Rockport IN 47635.

Micro Users Tax Booklet 78 shows how to slash taxes with deductions for micro. Many illustrations and examples. $4.99. MicroTax 78 completes 1040, schds A, B, C, D and others in minutes. Eliminates errors, missed deductions, and saves time. 16K TRS-80 Level II. $9.99. Save $2; order both for $12.98. Check or MO: George Clisham, PO Box 1172, Duxbury MA 02332.

SOL 20 Software. Excellent applications in home finance, education, games, etc. Manual, cassette included. Low cost, guaranteed. Catalogue, $1. G. Newby, 299 Dawlish Ave., Toronto Ontario Canada M4N 1J6.

TRS-80 software on cassette or disk: mail, inventory, word processor, data base, stock, check, statistic, sort. Poi Pow, 96 Dothan St., Arlington MA 02174.

TRS-80 Level II programs on cassette. 4K: Lunar/Martian Lander with 3 degrees of freedom ($8). 16K: Battle Klingons, Romulans with Star Trek ($10); you name teams in College/Pro Football ($12); shoot down attacking aircraft in Carrier Battle ($6). R. Fitzgerald, 3338 Sheila Lane #292, Dallas TX 75220.

Micropolis software special: Investment analysis package, $30; Home/family package, $30; Game package, $15; Assembly language prog. course for beginners, $30; Raw accts.-receivable program ready to personalize, $95. Order now. Phil Bell, 813 MacArthur Dr., Urbana IL 61801.

---

*Books of computer games in BASIC. ENIGMAS-1 ($8): Gone Fishing, Concentration, Starship, Craps, Slot-Machine, Sherlock Holmes, Tank Attack. ENIGMAS-2 ($8): Number Guess, Mortar Battle, In-Between, Shell Game, Safari, Starship-2, Dice Roll, Puzzle. *Catalog and test program Frog Race, $1. Available in Standard BASIC, SWTP 8K BASIC or Radio Shack TRS-80 BASIC. Please specify. B. Erickson, PO Box 11099, Chicago IL 60611.

TRS-80 Owners. I am interested in surveying TRS-80 users. Write: Professor Bill Parks, Chase-203, State University College, 1300 Elmwood Ave., Buffalo NY 14222.

CIRCLE CHESS instruction book by Alphonso, used in Circle Chess software development, $5. Stanonis, Box 63, Des Plaines IL 60017.

Educational games? SOLOMON HOUSE offers the Game of the States. PET graphics help make learning the names, spelling and geography of the states of the union fun for all ages. Information and list of available programs free. For Game of the States program tape and documentation send $4 to S. M. Solomon, 276 Buckingham Ave., Milford CT 06460.

For Sale: Imsai 8080, up and running. Too powerful for my needs. Extended MB, 3P&S, PROM/RAM start up, Tarbell intf., 32K memory, Extended 12K BASIC, all cables, manuals, software, etc. Also IBM Selectric I/O writer, as is. M. B. Hogan, 150-55 87 Ave., Jamaica NY 11432. 739-7031, 754-6442 (9:30-5:30). Price: $1999. With ASR-33 add $700.

For Sale: Heath H8, 16K, serial I/O bd, video terminal, cass. rec., all software. Running perfectly. $1350. Dr. Jay Wechsler. (609) 234-2175.

For Sale: Two SWTP 4K memory boards, $55 ea. SWTP CT-1024 TVT with 16 × 64 mod, $95. R. Stafford, 6799 McVey Blvd., Worthington OH 43085.

Honeywell 7600 Magnetic Tape System. 14 × 14 FM medium band records and reproduces. Make offer. D. Armstrong, 1584 Kingman Ave., San Jose CA 95128. (408) 374-4992.

Apple II Programs: Hi-res. drawing program (draws and saves any shape), 16 & 20K, $12; 16-color LIFE game, $6.50; astrology program (ephemeris needed), $10; number converter (bases 2-16), $5; graphics pak, $3. Send for current list. M. Klassen, 140 Willow Ave. #2, Fairfax CA 94930.

TRS-80 Level II users—Tired of searching through your TRS-80 Level II manual for a certain program command? My comprehensive index lets you find out what you need to know fast! Send $3.75 and stamped, self-addressed long envelope to Steven Benda, 503 East High St., Toledo IA 52342.

TRS-80 Level II, 16K. Instant keyboard graphics. Mach lang, less than 1K. Use in any BASIC II prog to graphically display 96 ASCII chars, magnify up to 6X. Up and lowercase & all spec char. Incl instr & 4 sample BASIC programs on tape. $12.95. To: Ed Warshawer, Box 438, Oakhurst NJ 07755.

TRS-80 Level I, 4K—5 pop games. Scoring and orig. graphics—all on 1 cassette, $7. C. Zalnerunas, 3034 W. Columbus Ave., Chicago IL 60652.

Heathkit ET3400 Microcomputer/Trainer—Motorola 6800 MPU, hex keyboard, readout, monitor, etc. Used with course EE3401 or as a single board computer. Like new, $149. V. M. Faulkner, RR2, Box 294A, Yorktown IN 47396. (317) 289-4138.

I wish to make contact with any reader or other interested persons on the subject of trend analysis in both commodities and stocks. I have many years of successful experience and wish to exchange ideas, methodology, programs, etc. Kindly contact Ed Tenberg, Box 407, Little Neck NY 11363.

For Sale: SWTP CT-1024 with all options, completely functional in Plexiglass case, full documentation & extra keyboard mount: $225, or best offer. MP-A processor board missing MC6800, 6810, MC1411, 8097 & 8098. Assembled w/MIKBUG & sockets in excellent condition: $60. Call or write: Kevin Wells, 712 N. Palm Dr., Beverly Hills CA 90210. (213) 271-1606.

H8 Owners: Fill and dump memory from your terminal in hex (instead of octal) using "Hexmem." Uses first K of user RAM, including stack. Easy to read format. $10 gets you cassette with binary, source listing and instructions. Specify del (7FH) or bs (08H) for typo correction. Patrick Swayne, 290 Springdale, Sebastopol CA 95472.

Multiprogramming Operating System for 6800 uP. System operates in conjunction with interrupt timer & maintains time of day. Send $10 for documentation & source code. Ed Joyce, 4603 Lyceum, San Antonio TX 78229.

CP-4 Control Data 9320 High Speed Printer, good condition with manuals, $175. CP-4 Remote Terminal Console with controls and PS, $75. R. Ackerman, 2 Hillside St., Red Bank NJ 07701.

Quality TRS-80 Programs. Business, Educational, Games, Ham Radio, Bridge, Inventory, mailing list, Accounting, disassembler, many more. Send SASE for information and complete list to: Sandy Stange, Box 145, Lithonia GA 30058.

Rockwell XPO-1 Development system for PPS-4/1 uPs. With optional assembler, p.s., 1K RAM. Never used. $450. G. Hyman, 22 Cross Hill Ave., Yonkers NY 10703. (914) 968-9700.

KIM Users: Powerful Console Input Monitor (CIM) lets you enter, edit and execute programs from ASCII encoded kybd on basic KIM. CIM modes are: Address, Program, Text, Data, Increment, Decrement and Execute. CIM was developed by an experienced EE and is fully supported for future expansion. Complete user's manual includes interfacing ASCII kybd to KIM. $16.50 ppd. Add $5 for CIM cassette. E. R. Kittlaus, Box 2175, Seal Beach CA 90740.

For sale: ASR-33 TTY with modem and stand, $650. Back issues of *Kilobaud, Byte,* others. Send SASE for complete list. Lenny Heath, Box 12545, Raleigh NC 27605. (919) 876-4168.

Quality Apple II Software—cassette & doc. —Disk-compatible catalog prog. (e.g., author-title index or disk prog. index) can fetch prog. names from D$CATALOG, $10; Auto Save/Restore disk to backup tape, $5; Graphic Blackjack, paddle-input, auto-play mode, $10. Any 2/$15. All 3/$18. George W. Lee, 18803 S. Christina Ave., Cerritos CA 90701.

TRS-80 quality Word Processor. Helps prepare letters, memos, documents and even a book. Level II, $25. DOS, $35. Poi Pow, 96 Dothan St., Arlington MA 02174.

TRS-80: Level II TLEG relocates your copy of T-BUG. Cassettes: $5.95. Super TLEGS moves with T-BUG so you can relocate at any time to any part of RAM. Cassette: $6.95. Either makes T-BUG a resident Z-80 monitor! Examine, modify formerly coincident programs. 1000s of uses. Allen Gelder, 5914 California St., San Francisco CA 94121.

Psychiatry/Clinical Psychology hardware, program library, current literature, reviews, etc. Bimonthly newsletter MICRO-PSYCH, 26 Trumbull St., New Haven CT 06511. $12/yr.

Netronics Elf II, giant board, 4K, power supply, keyboard, modulator, tiny BASIC, extras. $399. M. Otis, 1710 16th Ave., SE, Aberdeen SD 57401.

Wanted: *Kilobaud* #14 February 1978. Will pay $10 for a copy of issue #14 in good condition. William L. Schrader, K5NM 1409 Espanola N.E., Albuquerque NM 87110.

PET Owners: Send for free catalog of PET software. C.M.S, 5115 Menefee Drive, Dallas TX 75227.

SYM-1 SYM-1 SYM-1 SYM-1 SYM-1
New in original box with warranty, $220. Bill C., 13709 Peyton, Dallas TX 75240.
SYM-1 SYM-1 SYM-1 SYM-1 SYM-1

TRS-80 Users: Now play Kraps and Bakarat on 4K Level I or Level II system (specify). Guaranteed. Bonus game included. On cassette, $5.50. K. S. Schlaht, PO Box 2175, Seal Beach CA 90740.

PET 2001-8K. Hardly used. In warranty up to 2/10/79. With 4 cassettes. For $699 only. Call Salah at 914-226-4205. Evenings.

Apple II SSTV software. Display slow-scan TV pictures in hi-res graphics. Software demodulated—just connect receiver audio to Apple—no hardware needed! Req. 16K. On cassette with test pictures, $18. Listing only, $12. C. H. Galfo, 602 Orange St., Charlottesville VA 22901.

1802 Elf II Users—Play Battlestar Galactica, Slot Machine, Craps, Blackjack. $10 for all four games on a single cassette with documentation. Each program requires 2K RAM addressed 0000-07FF. Cylons dodge and attack, wheels spin, dice tumble. Send money order to: John Beringer, 2500 Teddy Dr., Apt. 29, Las Vegas NV 89102.

KB CALENDAR

Blacksburg VA

A new and expanded series of four 3-day hands-on workshops on 8080/8085 design, microcomputer interfacing, software design and digital electronics is being given by the authors of the popular Bugbooks. Participants have the option of retaining equipment used in these courses. Dates are March 19 to 28, 1979. For more information, contact Dr. Linda Leffel, C.E.C., Virginia Polytechnic Institute and State University, Blacksburg VA 24061, (703) 961-5241.

Boston MA

Coming Friday, Saturday and Sunday, April 6, 7 and 8, The Northeast Personal and Business Computer Show, Hynes Auditorium, Prudential Center, Boston MA. For more information, call or write: Show Manager, Northeast Personal and Business Computer Show, PO Box 678, Brookline Village MA 02147, (617) 522-4467.

Contest!

Our "best article of the month" contest continues. Turn to the reader-service card in the back of the magazine to vote.

Voted best article in the November 1978 issue was "Hey, Kids! It's 'Mickey Modem'!" by Stephen Gibson (p. 52). Congratulations, Steve.

Next month, we'll announce the best article of the year.

CORRECTIONS

The last sentence of the last full paragraph on p. 65 of "Spelling Bee" (December 1978) by David Moody should read: "The unnumbered lines following 5700, 5800, 6500 and 8600. . . ." not "5400, 5800, 6500 and 6800."

William Colsher tells us that you should add 275 M = M + 1 and that you should change line 340 to read GOTO 120 in "Lucas' Puzzle" (November 1978, p. 98).

Jim Gross writes: "I was flattered to see my article, 'Loop the Loop,' in the November 1978 issue. I am now ashamed to point out that the code in Example 7, page 115, is incorrect. Here is a listing of the program as it should be, and I apologize for any inconvenience the error may have caused."

```
 2  REMARK     SQUARE ROOTS BY NEWTON'S METHOD
 5  PRINT "WHAT VALUE DO YOU WANT THE SQUARE ROOT OF";
10  INPUT A
15  LET G = A
20  LET X = .5*(G + A/G)
21  PRINT "X IS NOW ";X
23  IF ABS (G/X – 1) < .00001 THEN 30
25  LET G = X
27  GOTO 20
30  PRINT "MY BEST ESTIMATE OF THE SQUARE ROOT OF ";A;" IS ";X
31  PRINT
32  GOTO 5
OK
```

From Belgium, Bruce Turrie writes to tell of some problems with his "Swords and Sorcery" program (August 1978, p. 54).

"Statement 840 does nothing; the lines that used to be between 840 and 890 were removed during debugging. However, since other statements do branch to 840, you must be careful about removing it; perhaps 840 REM would be best. This is a good example of what can happen to your code without hard copy.

"Line 3115 should be C2 = C2 – 1.

"I find that the line which should be 8350 is shown as 8340; that is, it should be 8350 PRINT "YOU'RE HIT!!!"

"Digits = 1 sets the number of digits to be printed to the right of the decimal point to one. This is the minimum for my BASIC, though I would have preferred integers."

Stuart Rowland writes again: "Thank you for printing my letter in the November 1978 *Kilobaud* (p. 22). Unfortunately, the published version contained some errors that would make it unintelligible to a novice. In order to clarify the algorithms, I'd like to rewrite them in a more English-like language.

"Let b(n), b(n – 1), . . . , b(1) be the bits in an n-bit number.

"Binary to Gray code algorithm: FOR i: = 2 TO n DO IF b(i) = 1 THEN complement b(i – 1).

"Gray to binary code algorithm: FOR i: = n DOWNTO 2 DO IF b(i) = 1 THEN complement b(i – 1).

"Another error of lesser importance occurs on the top line of column 3. The word transistor should be transition."

✓ Reader Service—see page 179

THE BEST FOR BOTH WORLDS

Factory prime electronic equipment for both the discerning hobbyist and the professional

The New Hobby World Catalog

Your source for factory prime, professional quality equipment. Computers, add-on boards, IC's, sockets, resistors, supplies, tools, test equipment, books, and more. Shop your buy list at Hobby World. You'll find what you want, and at a solid savings.

This month's specials.

16K MEMORY ADD-ON FOR TRS-80, APPLE, & EXIDY SORCERER
With RAMs, jumpers, and instructions. Hobby World price only **$98.** (Specify when ordering)

THE NEW ELENCO 3½ DIGIT SOLID STATE MULTIMETER
Tops in performance! Measures resistance to .01 ohms voltage to 100 μV, current to 1 μA. Our price only **$75** assembled & tested. **$60 kit.**

16K STATIC RAM KIT, MODEL XVI.
IEEE S-100 compatible, requires only +5 Volts. 450 ns, fully buffered. Bank switching capability, Phantom Line, wait state, addressable in 4K blocks. New from CALIFORNIA COMPUTER SYSTEMS! Our Price only **$265** kit, **$27** Bare Board.

C-10 DATA CASSETTES
Perfect for TRS-80 and all recorders operating under KC or Tarbell standards. With positive tape path control. Price only **$2** each! 10 for **$17.50.**

PROM PROGRAMMER
New from OAE! Pocket size, Read, program, verify, duplicate. Plugs into any read only PROM socket. With features like Zero insertion force socket, cermet trimmers, and 5′ of connecting ribbon cable. Hobby World price only **$275.**
Type # 1087 programs 2708's and TMS2716's
Type # 1088 programs Intel 2708's.

PORTABLE PAPER TAPE READER
No moving parts, interfaces with all microprocessors via an 8 bit parallel port. Complete with precision optical sensor array, high speed data buffers, all required handshake logic, 4 status LED's, flat ribbon interface cable and complete instructions. Hobby World price only **$74.00** kit, **$87.50** assembled and tested.

TRS-80 SOFTWARE
Order by Catalog Number.

CAT NO.	TITLE	LEVEL	PRICE
1093	Sargon: Chess Program	2	**$19.95**
1040	Galactic Blockade Runner	½	**9.95**
1041	Star Trek III	2	**14.95**
1043	Small Business Bookkeeping	½	**14.95**
1045	Bingo	½	**5.95**
1036	SCI-FI Game Sampler	½	**5.95**
1037	The Magic Isle	½	**5.95**
1038	Inventory (Modular)	½	**19.95**
1047	Othello	½	**5.95**
1048	Machine Language Monitor	2	**23.95**
1049	Micro Text Editor	2	**5.95**
1050	100 Programs, Home, Business, and Games	2	**49.50**
1051	Daily Biorhythm Program	½	**5.95**

Hobby World
19355 Business Center Drive #6
Northridge, CA 91324
SEND ME A CATALOG! ✓ **H25**

Name

Address

City State

Order by mail. Just write. Or order by phone. Toll free outside California.

TELEPHONE ORDERS
Inside Calif: 213 886-9200
Outside Calif: 800 423-5387

BANKAMERICARD VISA

master charge THE INTERBANK CARD

SEND ME THE FOLLOWING SPECIALS!

Item	Quantity	Price
	Postage/Handling	$1.50

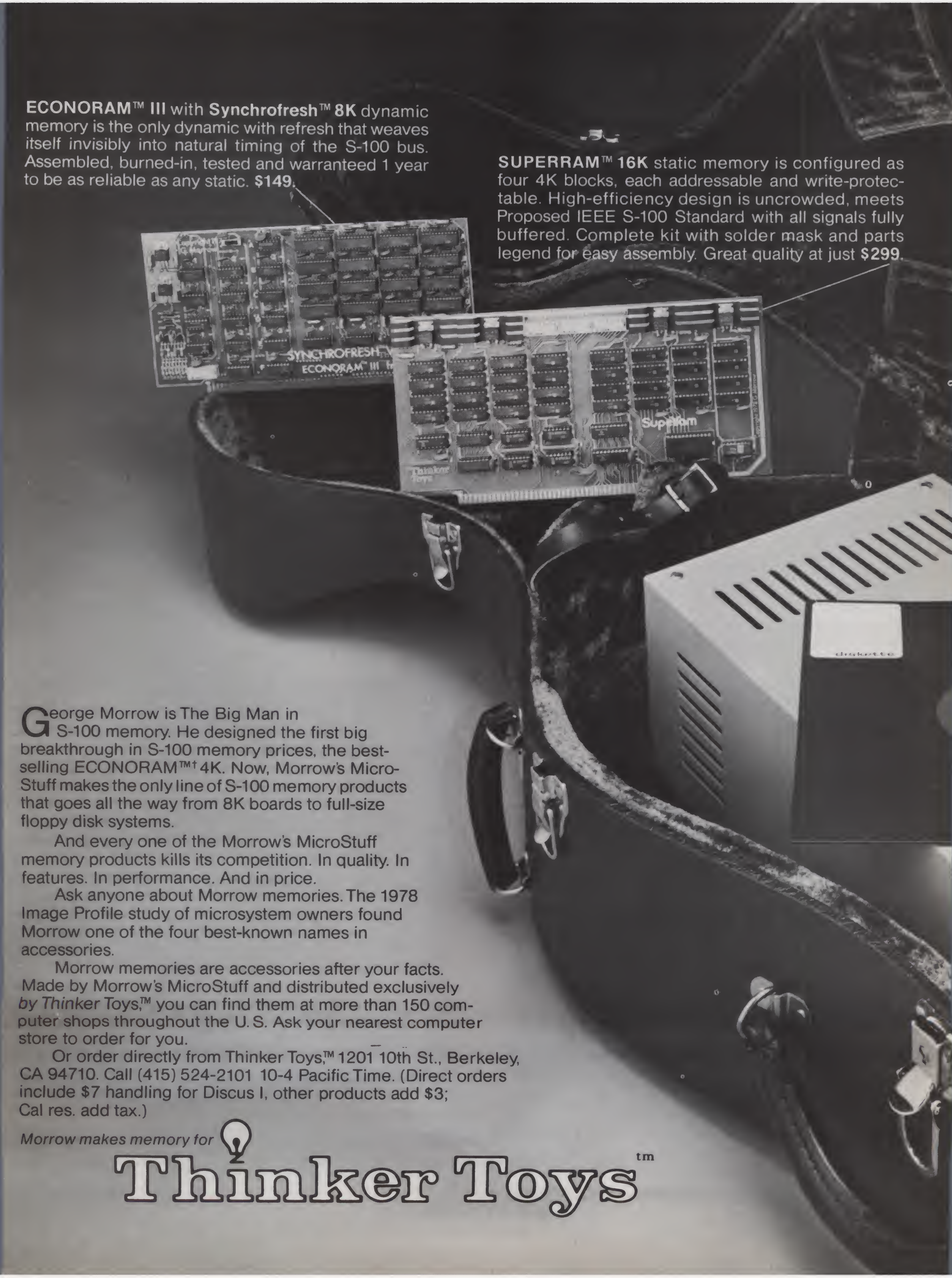
Satisfaction 100% Guaranteed

California Residents add 6% Sales Tax
Charge My MC #
VISA

Signature

Expiration

✓ *Reader Service—see page 179*


ECONORAM™ III with **Synchrofresh™ 8K** dynamic memory is the only dynamic with refresh that weaves itself invisibly into natural timing of the S-100 bus. Assembled, burned-in, tested and warranteed 1 year to be as reliable as any static. **$149.**
SUPERRAM™ 16K static memory is configured as four 4K blocks, each addressable and write-protectable. High-efficiency design is uncrowded, meets Proposed IEEE S-100 Standard with all signals fully buffered. Complete kit with solder mask and parts legend for easy assembly. Great quality at just **$299.**
SYNCHROFRESH
ECONORAM III
SuperRam
Thinker Toys
George Morrow is The Big Man in S-100 memory. He designed the first big breakthrough in S-100 memory prices, the best-selling ECONORAM™† 4K. Now, Morrow's MicroStuff makes the only line of S-100 memory products that goes all the way from 8K boards to full-size floppy disk systems.
And every one of the Morrow's MicroStuff memory products kills its competition. In quality. In features. In performance. And in price.
Ask anyone about Morrow memories. The 1978 Image Profile study of microsystem owners found Morrow one of the four best-known names in accessories.
Morrow memories are accessories after your facts. Made by Morrow's MicroStuff and distributed exclusively *by Thinker Toys,™* you can find them at more than 150 computer shops throughout the U.S. Ask your nearest computer store to order for you.
Or order directly from Thinker Toys,™ 1201 10th St., Berkeley, CA 94710. Call (415) 524-2101 10-4 Pacific Time. (Direct orders include $7 handling for Discus I, other products add $3; Cal res. add tax.)
Morrow makes memory for
Thinker Toys™

JADE Computer Products

★ DISK DRIVES ★

B51 - 5¼" **$295.00**

by Micro Peripherals, Inc. Operates in either single density (125KB, unformatted) or double density (250KB, unformatted) modes, up to 40 tracks, with a track-to-track access time of only 5 ms.

SA801R **$495.00**

by Shugart. Single-sided 8" floppy disk drive.

8" DISK DRIVE **$395.00**

by GSI/Siemans. Direct equivalent of Shugart 801R.

DM 2700-S **$750.00**

Includes SA801R, 10"x10"x16" cabinet, power supply, data cable, fan, AC line filter.

- TWO SIEMENS/GSI 8" FLOPPY DRIVES
- POWER SUPPLY FOR ABOVE
- JADE/TARBELL DISK CONTROL KIT (S100)
- CP/M OPERATING SYSTEM WITH BASIC-E
- PACKAGE OF 10 BLANK 8" DISKETTES

Price if purchased separately: $1192.50

JADE SPECIAL PACKAGE DEAL: $1050.00

VERBATIM T.M. FLOPPY DISKS

5¼ in. Minidiskettes

Soft sector, 10 sector, or 16 sector

$4.40 each or box of 10 for $40.00

8 in. Standard Floppy Disks

Soft Sector

$4.75 each—10 for $42.50

FLOPPY DISK INTERFACE

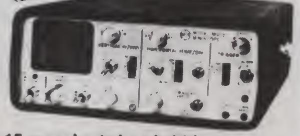
JADE FLOPPY DISK (Tarbell board)

Kit	$175.00
Assembled & Tested	$250.00

S.D. Computer Products "Versa Floppy"

Kit	$159.95
Assembled & Tested	$189.95

MINISCOPES



- 15 megahertz bandwidth.
- External and internal trigger.
- Time Base - 0.1 microseconds to 0.5 Sec/div - 21 settings.
- Battery or line operation.
- Automatic and line sync modes.
- Power consumption less than 15W
- Verticle Gain - 0.01 to 50 volts/div-12 settings.
- Weight is only 3 pounds.

MS-15	$ 318
MS-215 Dual Trace Version	$ 435

With Rechargeable Batteries & Charger Unit

10 to 1, 10 meg probe	$ 27.
Leather carrying case	$ 45.

LOGIC PROBES

MODEL LP-1

Hand-held logic probe provides instant reading of logic levels for DTL, HTL, TTL, or CMOS. **Input Impedance:** 100,000 ohms. **Minimum Detectable Pulse:** 50 ns. **Maximum Input Signal (Frequency):** 10 MHz. **Pulse Detector (LED):** High speed train or single event. **Pulse Memory:** Pulse or level transition detected and stored

CSC Model LP-1 Logic Probe Net Each **$44.95**

MODEL LP-3

High speed logic probe. Captures pulses as short as 10 ns. **Input Impedance:** 500,000 ohms. **Minimum Detectable Pulse:** 10 ns. **Maximum Input Signal (Frequency):** 50 MHz. **Pulse Detector (LED):** High speed train or single event. **Pulse Memory:** Pulse or level transition detected and stored

CSC Model LP-3 Logic Probe Net Each **$69.95**

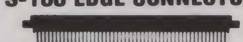
3-LEVEL GOLD WIRE WRAP SOCKETS

14 PIN	39¢ each
16 PIN	43¢ each

100 for $30.00

Sockets are end and side stackable, closed entry.

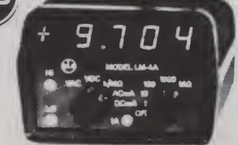
GOLD PLATED S-100 EDGE CONNECTORS



Soldertail $3.25 each
10 for $30.00

Wire Wrap $4.50 each
10 for $40.00

METERS



Rechargeable batteries and charger
Measures DC Volts, AC Volts, Ohms and Current
Automatic polarity, decimal and overload indication
No zero adjustment and no full-scale ohms adjust
Battery-operated — NICad batteries; also AC line operation.
Large LED display for easy reading without interpolation
Size: 1.9"H x 2.7"W x 4"D
Parts & labor guaranteed 1 year

Tilt stand option .... $ 3.50
Leather case $20.00

LM3A 3 dig 1% DC $134.00
LM3.5A 3½ dig .5% DC ..$ 158.50
LM40A 4 dig .1% DC.....$ 209.00
LM4A 4 dig .03% DC .... $ 250.00

S-100 MOTHER BOARDS

JADE 6-SLOT

Kit	$41.95
Assembled & Tested	$56.95
Bare Board	$24.95

9–SLOT "LITTLE MOTHER"

Kit	$85.00
Assembled & Tested	$99.00
Bare Board	$35.00

13–SLOT "QUIET MOTHER"

Kit	$95.00
Assm. & Tested	$110.00
Bare Board	$40.00

22–SLOT "STREAKER"

Assm. & Tested	$149.00

JADE VIDEO INTERFACE

S-100 Compatible Serial Interface with Sockets Included.

Kit	$117.95
Assembled & Tested	$159.95
Bare Board w/manual	$ 35.00

Z80A SPECIAL

4MHz Zilog CPU Chip

☆ $14.95 ☆

Zilog

SEE OPPOSITE PAGE FOR ORDERING INSTRUCTIONS.

TV-1

Convert your T.V. set into a Video Monitor

Kit $8.95

ELECTRONIC SYSTEMS

"KANSAS CITY STANDARD" TAPE INTERFACE

Part No. 111

Board $7.60; with parts $27.50

RS-232/TTL INTERFACE

Part No. 232

Converts TTL to RS-232 and RS-232 to TTL

Board only $4.50; with parts $7.00

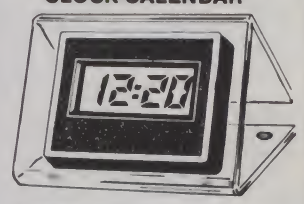
RS-232/TTY INTERFACE

Part No. 600

Converts RS-232 to 20mA current loop, and 20mA current loop to RS-232

Board only $4.50; with parts $7.00

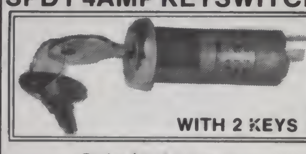
LIQUID CRYSTAL DIGITAL CLOCK-CALENDAR



- For Auto, Home, Office
- Small in size (2x2½x½)
- Push button for seconds release for date.
- Clocks mount anywhere with either 3M double-sided tape or VELCRO, included.
- 2 MODELS AVAILABLE.
  LCD-101, portable model runs on self-contained batteries for better than a year.
  LCD-102, runs on 12 Volt system and is back lighted.
- LCD-101 or LCD-102 your choice **$34.95 ea**
- Clear desk stand for **$2.00**

Protect Your Hardware From Unauthorized Use:

SPDT 4AMP KEYSWITCH



WITH 2 KEYS

Only $2.95 Each

JADE PARALLEL/SERIAL INTERFACE

S-100 compatible, 2 serial I/O ports, 1 parallel I/O.

Kit JG-P/S	$124.95
Assembled & Tested: JG-P/SA	$179.95
Bare Board w/Manual	$ 30.00

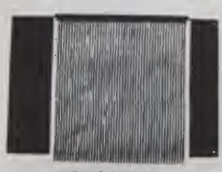
EXPANDOR'S BLACK BOX PRINTER

This 64-character ASCII impact printer with 80-column capability is portable and uses standard 8½" paper and regular typewriter ribbon. Base, cover and parallel interface are included. Assembled and complete with manual and documentation. **only $470.00**

(90 day manufacturer's warranty)

TRS-80 Interface Cable for Black Box Printer with mating connectors: **$48.00**
(must be used with expansion module, +8v/1 amp power supply required.)

Power Supply for TRS–80/Black Box Printer **$49.00**



3690-12
CARD EXTENDER

Card Extender has 100 contacts 50 per side on .125 centers. Attached connector is compatible with S-100 Bus Systems $25.00.
3690 6.5" 22/4 pin. 156 ctrs. Extenders $12.00

Gen. Purpose D.I.P. Boards with Bus Pattern for Solder or Wire Wrap. Epoxy Glass 1/16" 44 pin con. spaced .156

3677 9.6" x 4.5" $10.90
3677-2 6.5" x 4.5" $9.74

Vector Plugboards

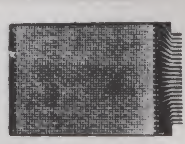
8800V

Universal Microcomputer/processor plugboard use with S-100 bus. Complete with heat sink & hardware 5.3" x 10" x 1/16

8801-1

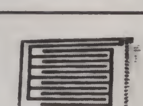
Same as 8800V except plain less power buses & heat sink

	1-4	5-9	10-24
8800V	19 95	17.95	15 96
8801-1	14 95	13 46	11 96



P pattern plugboards for IC's Epoxy Glass 1/16" 44 pin con. spaced .156

3662 6.5" x 4.5" $7.65
3662-2 9.6" x 4.5" $11.45



Hi-Density Dual-In-Line Plugboard for Wire Wrap with Power & Grd. Bus Epoxy Glass 1/16" 44 pin con. spaced .156

3682 9.6" x 4.5" $10.97
3682-2 6.5" x 4.5" $9.81

PLACE ORDERS TOLL FREE:

800/421-5809 Continental U.S.
800/262-1710 Inside California

MICROPROCESSORS

F8	16.95
Z80 (2MHz)	20.00
Z80A (4MHz)	25.00
CDP1802CD	17.95
AM2901	20.00
6502	11.95
6800	16.95
6802	25.00
8008-1	12.95
8035	20.00
8035-8	21.00
8080A	10.00
8085	23.00
TMS9900TL	49.95

8080A SUPPORT DEVICES

8212	2.90
8214	4.65
8216	2.75
8224 (2MHz)	4.30
8224-4 (4MHz)	9.95
8226	2.75
8228	6.40
8238	6.40
8243	8.00
8251	7.50
8253	20.00
8255	8.50
8257	20.00
8259	20.00
8275	75.00
8279	18.50

6800 PRODUCT

6810P	4.00
6820P	6.60
6821P	6.60
6828P	11.25
6834P	16.95
6844L	29.95
6845L	29.95
6846L1	35.00
6846P1	30.00
6850P	8.65
6852P	11.00
6860P	9.25
6862P	12.00
6871P	28.75
6875P	8.75
6880P	2.50

KIM SUPPORT DEVICES

6102	8.00
6502	11.95
6520	10.00
6522	9.25
6530-002	15.95
6530-003	15.95
6530-004	15.95
6530-005	15.95
6532	17.95

USRT

S2350	10.95

UARTS

AY5-1013A	5.25
AY5-1014A	8.25
TR1602B	5.25
TMS6011	5.95
IM6402	9.00
IM6403	9.00

BAUD RATE GENERATORS

MC14411	10.00
14411 Crystal	4.95

CHARACTER GENERATORS

2513 Upper (-12±5)	6.75
2513 Lower (-12 ±5)	6.75
2513 Upper (5volt)	9.75
2513 Lower (5 volt)	10.95
MCM6571 - Up Scan	10.95
MCM6571A - Down Scan	10.95

FLOPPY DISC CONTROLLER

1771B01	39.95
1791	49.95

KEYBOARD CHIPS

AY5-2376	13.75
AY5-3600	13.75
MM5740	18.00

PROM S

1702A	8.00
2708	9.95
2716(5+12)TI	25.00
2716(5v) INTL	60.00
2758(5v)	23.40

DYNAMIC RAMS

416D/4116 (250ns)	12.50
2104/4096	4.00
2107B-4	3.95
TMS4027/4096 (300ns)	4.00
MM5270	4.50
MM5280	3.60

STATIC RAMS	1–24	25-99
21L02 (450ns)	1.50	1.20
21L02 (250ns)	1.75	1.50
2101-1	2.95	2.60
2111-1	3.25	3.00
2112-1	2.95	2.65

Rockwell AIM-65: The Head-Start in Microcomputers

A KIM-1 compatible machine with on-board printer and a real keyboard!

$375.00 w/1K RAM
$450.00 w/4K RAM

4K assembler/editor in ROM:	$ 80.00
8K BASIC in ROM:	$100.00
Power supply:	$ 59.95
Case for AIM-65:	$ 49.95

Special Package Price: $599.00
AIM-65 (4K), Power Supply, Case, and 8K BASIC ROM

TRS-80 apple II

MEMORY EXPANSION KITS
4116's
8 for $85.00
(16K x 1, 200ns)
Includes dip plugs and instructions

★TRS-80 Kit ★
(16K x 1, 300ns)
includes connectors and instructions
$90.00

THE KIM-1

$179

Low price includes KIM-1 Module, monitor programs stored in 2048 ROM Bytes, User Manual, wall size Schematic, Hardware Manual, Programming Manual, Programmers Reference Card, Keyboard/Display

JADE 8080A

with full documentation

Kit	$100.00
Assembled & Tested	$149.95
Bare Board	$ 30.00

"IMSAI"-TYPE CARD GUIDE SPECIAL:

Regular Price 30¢ each

SPECIAL: 10 for $1.00!

THE SYM-1 $245

6502 - based single board computer with keyboard/display, KIM-1 hardware compatible, complete documentation.

JADE Z80

with provisions for ONBOARD 2708 and POWER ON JUMP

2MHz

Kit	$135.00
Assembled & Tested	$185.00

4 MHz

Kit	$149.95
Assembled & Tested	$199.95
Bare Board	$ 35.00

Sale!

DYNAMIC RAM BOARDS EXPANDABLE TO 64K

32K VERSION • KITS

Uses 4115 (8Kx1, 250ns) Dynamic RAM's, can be expanded in 8K increments up to 32K:

8K	$159.00
16K	$199.00
24K	$249.00
32K	$299.00

64K VERSION • KITS

Uses 4116 (16Kx1, 200ns) Dynamic RAM's, can be expanded in 16K increments up to 64K:

16K	$249.00
32K	$369.00
48K	$475.00
64K	$575.00

KIMSI

INTERFACE/MOTHERBOARD
Makes S-100 cards plug-in compatible with KIM!

Kit	$125.00
Assembled & Tested	$165.00

BETSI

INTERFACE/MOTHERBOARD
Makes S-100 cards plug-in compatible with PET!

Kit	$119.00
Assembled & Tested	$159.00

★ STATIC RAM ★ SPECIALS

2114's, low power (1024x4)

	1-15	16-99	100 +
450ns	8.00	6.95	5.50
250ns	9.00	8.00	6.50

TMS4044/MM5257, low power

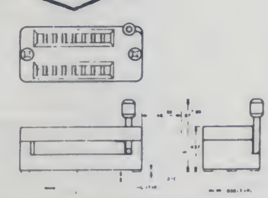
450ns	8.00	7.50	6.50
250ns	9.95	8.75	8.00

4200A (4Kx1, 200ns)

	9.95	8.50	8.00

ZIP DIP® II Socket

TEXTOOL



This new type of zero insertion pressure dual in-line package socket (ZIP DIP II) is perfectly suited for both hand test and burn-in requirements.

The ZIP DIP II socket has been designed for the utmost simplicity in its mechanical action. Coupled with a thoughtful system of ramps and bevels to guide the device leads into the contacts results in a socket, into which, the device can literally be dropped. With the flip of a locking lever the socket is ready to operate with exceptionally good electrical contact. Flip the lever again and the device may be extracted with zero pressure being exerted on the leads by the socket contacts.

PRICES:

16 pin Zip Dip II	$5.50
24 pin Zip Dip II	$7.50
40 pin Zip Dip II	$10.25

EPROM BOARD KITS

EPM-1 (uses up to 4K of 1702)	$59.95
JG8/16 (uses 2708 or 2716)	$69.95

STATIC RAM BOARDS

JADE 8K

Kits:	450ns	$125.95
	250ns	$149.75
Assembled & Tested:	450ns	$139.75
	250ns	$169.75
Bare Board:		$ 25.00

16K — Uses 2114's (low power)
Assembled & Tested:

RAM 16 (250ns)	$375.00
RAM 16B (450ns)	$325.00

16K with memory management
Assembled and Tested:

RAM 65 (250ns)	$390.00
RAM 65B (450ns)	$350.00

32K Static
Assembled & Tested:

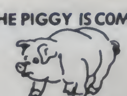
250ns	$795.00
450ns	$725.00
250ns Kit	$575.00

JADE Computer Products

✓J6

4901 W. ROSECRANS AVENUE
Department "K"
HAWTHORNE, CALIF. 90250

THE PIGGY IS COMING!

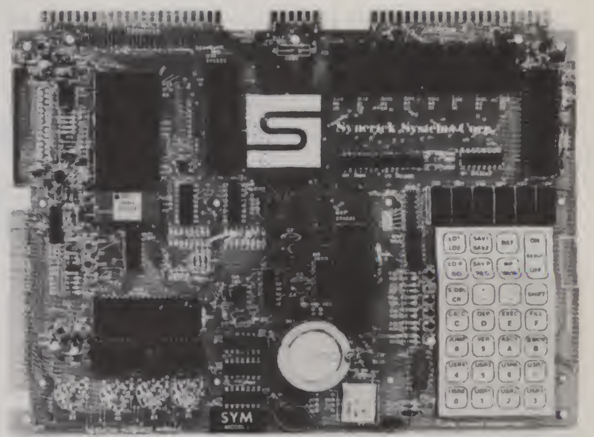


Cash, checks, money orders, and credit cards accepted. Add freight charge of $2.50 for orders under 10 lbs. and $1.00 service charge for orders under $10.00. Add 6% sales tax on all parts delivered in California. Discounts available at OEM quantities.

WRITE FOR OUR FREE CATALOG
All prices subject to change without notice.

SYM-1, 6502-BASED MICROCOMPUTER

- FULLY-ASSEMBLED AND COMPLETELY INTEGRATED SYSTEM that's ready-to-use
- ALL LSI IC'S ARE IN SOCKETS
- 28 DOUBLE-FUNCTION KEYPAD INCLUDING UP TO 24 "SPECIAL" FUNCTIONS
- EASY-TO-VIEW 6-DIGIT HEX LED DISPLAY
- KIM-1* HARDWARE COMPATIBILITY
  The powerful 6502 8-Bit MICROPROCESSOR whose advanced architectural features have made it one of the largest selling "micros" on the market today.
- THREE ON-BOARD PROGRAMMABLE INTERVAL TIMERS available to the user, expandable to five on-board.
- 4K BYTE ROM RESIDENT MONITOR and Operating Programs.
- Single 5 Volt power supply is all that is required.
- 1K BYTES OF 2114 STATIC RAM onboard with sockets provided for immediate expansion to 4K bytes onboard, with total memory expansion to 65, 536 bytes.
- USER PROM/ROM: The system is equipped with 3 PROM/ROM expansion sockets for 2316/2332 ROMs or 2716 EPROMs
- ENHANCED SOFTWARE with simplified user interface
- STANDARD INTERFACES INCLUDE:
  —Audio Cassette Recorder Interface with Remote Control (Two modes: 135 Baud KIM-1* compatible, Hi-Speed 1500 Baud)
  —Full duplex 20mA Teletype Interface
  —System Expansion Bus Interface
  —TV Controller Board Interface
  —CRT Compatible Interface (RS-232)
- APPLICATION PORT: 15 Bi-directional TTL Lines for user applications with expansion capability for added lines
- EXPANSION PORT FOR ADD-ON MODULES (51 I/O Lines included in the basic system)
- SEPARATE POWER SUPPLY connector for easy disconnect of the d-c power
- AUDIBLE RESPONSE KEYPAD



Synertek has enhanced KIM-1* software as well as the hardware. Th software has simplified the user interface. The basic SYM-1 system i programmed in machine language. Monitor status is easily accessible and the monitor gives the keypad user the same full functional capabil ty of the TTY user. The SYM-1 has everything the KIM-1* has to offe plus so much more that we cannot begin to tell you here. So, if you wa to know more, the SYM-1 User Manual is available, separately.

SYM-1 Complete w/manuals	**$269.00**
SYM-1 User Manual Only	**7.00**
SYM-1 Expansion Kit	**75.00**

Expansion includes 3K of 2114 RAM chips and 1-6522 I/O chip.

SYM-1 Manuals: The well organized documentation package is com plete and easy-to-understand.

SYM-1 CAN GROW AS YOU GROW. Its the system to BUILD-ON. E pansion features that are soon to be offered:

8K Basic ROM	**$159.00**
TV Interface Board	**349.00**

QUALITY EXPANSION BOARDS DESIGNED SPECIFICALLY FOR KIM-1, SYM-1 & AIM 65

These boards are set up for use with a regulated power supply such as the one below, but, provisions have been made so that you can add onboard regulators for use with an unregulated power supply. But, because of unreliability, we do not recommend the use of onboard regulators. All I.C.'s are socketed for ease of maintenance. *All boards carry full 90-day warranty.*

All products that we manufacture are designed to meet or exceed industrial standards. All components are first qualtiy and meet full manufacturer's specifications. All this and an extended burn-in is done to reduce the normal percentage of field failures by up to 75%. To you, this means the chance of inconvenience and lost time due to a failure is very rare; but, if it should happen, we guarantee a turn-around time of less than forty-eight hours for repair.

Our money back guarantee: If, for any reason you wish to return any board that you have purchased directly from us within ten (10) days after receipt, complete, in original condition, and in original shipping carton; we will give you a complete credit or refund less a $10.00 restocking charge per board.

VAK-1 8-SLOT MOTHERBOARD

This motherboard uses the KIM-4* bus structure. It provides eight (8) expansion board sockets with rigid card cage. Separate jacks for audio cassette, TTY and power supply are provided. Fully buffered bus.

VAK-1 Motherboard **$129.00**

VAK-2/4 16K STATIC RAM BOARD

This board using 2114 RAMs is configured in two (2) separately addressable 8K blocks with individual write-protect switches.

VAK-2 16K RAM Board with only 8K of RAM (½ populated)	**$239.00**
VAK-3 Complete set of chips to expand above board to 16K	**$175.00**
VAK-4 Fully populated 16K RAM	**$379.00**

VAK-5 2708 EPROM PROGRAMMER

This board requires a +5 VDC and $\pm$12 VDC, but has a DC to DC multiplyer so there is no need for an additional power supply. A software is resident in on-board ROM, and has a zero-insertion socke

VAK-5 2708 EPROM Programmer **$269.00**

VAK-6 EPROM BOARD

This board will hold 8K of 2708 or 2758, or 16K of 2716 or 25 EPROMs. *EPROMs not included.*

VAK-6 EPROM Board **$129.00**

VAK-7 COMPLETE FLOPPY-DISK SYSTEM (Feb. '79)

VAK-8 PROTYPING BOARD

This board allows you to create your own interfaces to plug into t motherboard. Etched circuitry is provided for regulators, address a data bus drivers; with a large area for either wire-wrapped or soldere IC circuitry.

VAK-8 Protyping Board **$49.00**

POWER SUPPLIES

ALL POWER SUPPLIES are totally enclosed with grounded enclosures for safety, AC power cord, and carry a full 2-year warranty.

FULL SYSTEM POWER SUPPLY

This power supply will handle a microcomputer and up to 65K of our VAK-4 RAM. ADDITIONAL FEATURES ARE: Over voltage Protection on 5 volts, fused, AC on/off switch. Equivalent to units selling for $225.00 or more.

Provides +5 VDC @ 10 Amps & +12 VDC @ 1 Amp
VAK-EPS Power Supply **$125.00**

KIM-1* Custom P.S. provides 5 VDC @ 1.2 Amps and +12 VDC @ .1 Amps
KCP-1 Power Supply **$41.50**

SYM-1 Custom P.S. provides 5 VDC @ 1.4 Amps
VCP-1 Power Supply **$41.50**

*KIM is a product of MOS Technology

RNB ENTERPRISES INCORPORATED

2967 W. Fairmount Avenue
Phoenix AZ. 85017
(602)265-7564

master charge VISA

Prices in effect Nov. '

ADVANCED COMPUTER PRODUCTS

✓ A38

THE FIRST TO OFFER PRIME PRODUCTS TO THE HOBBYIST AT FAIR PRICES NOW LOWERS PRICES EVEN FURTHER!

1. Proven Quality Factory tested products only, no re-tests or fallouts. Guaranteed money back. We stand behind our products.

2. Same Day Shipment All prepaid orders with cashiers check, money order or charge card will be shipped same day as received.

STATIC RAM BOARDS

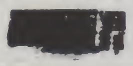
Now, just more than 1¹/₂¢ per bit!

S-100 32K (uses 2114)
ASSEMBLED 450ns. 639.95 | Kit 450ns. 539.95
250ns. 699.95 | 250ns. 599.95
Bare Board 49.95
Bare Board w/all parts less mem. 99.95

WMC 16K (uses 2114)
ASSEMBLED 450ns. 325.00 | KIT 450ns. 279.00
250ns. 375.00 | 250ns. 299.00
Bare Board 29.95

LOGOS I 8K
ASSEMBLED 450 ns. 149.95 | KIT 450ns. 125.95
250ns. 169.95 | 250ns. 149.95

Bare PC Board w/Data $21.95
Now over 1 year successful field experience
"Special Offer" Buy (4) 8K 450ns. Kits $117.00



Z-80/Z-80A/8080 CPU BOARD

★ On board 2708 ★ 2708 included (450ns.)
★ Power on jump ★ completely socketed
Assembled and tested $185.00
Kit $129.95
Bare PC Board .... $ 34.95
★ For 4MHz Speed Add $15.00
8080A Kit .... $ 99.95
8080A Assembled $149.95

ACOUSTIC COUPLERS • MODEMS

We now stock complete AJ line.
A242A 300 Baud Coupler $325.00
AJ243 LA 36 Coupler $295.00
ADAC242 Refurb. Coupler $275.00
AD342 Orig./Ans. Coupler $398.00
AJ1245 ½ Duplex 300/1200 Bd $795.00
AJ1234 Vadic Comp 1200 Bd Mod $895.00
AJ1255 1200 Bd. Vadic Orig/Ans. $895.00
Plus more Anderson Jacobson Products

EXIDY SORCERER

Complete expandable Z-80 based computer.
w/8K $895.00 w/16K $1150.00
w/32K $1395.00 stock
S-100 Expansion module $299.00

Hi PLOT LOW COST DIGITAL PLOTTER

IN STOCK

★ RS 232
★ Plot Size 7" x 10"
★ Digitizer Avail. Soon
★ High Resolution
★ 2.4"/sec Plot Speed

NEW List: $1085.00
OUR PRICE $999.00

SPECIAL KEYBOARD BUY WHILE THEY LAST

"Clare Pender 63 Key ASCII w/26 Pin & 34 Pin Output Conn." ... $54.95

BYTE USER 8K EPROM BOARD

★ Power on Jump ★ Reset Jump
Assembled & Tested $94.95
Kit $64.95
Bare PC Board $21.95
Special Offer: Buy 4 kits only $59.95 each
MR-8 8K w/1K Ram $99.50
MR-16 16K w/1K Ram $99.50
EPM-1 4K 1702 .... $59.95
EPM-2 2708 or 2716 Eprom $69.95
NOTE: 2708-6 only $5.95

IMS STATIC RAM BOARDS

★ Memory Mapping ★ Low Power
★ Phantom ★ 250 ns. or 450 ns.
Only available assembled & tested

	250 ns.	450 ns.
8K Static	$209.00	$189.00
16K Static	$449.00	$399.00
32K Static	$869.00	$819.00

TARBELL FLOPPY INTERFACE

★ S100 Compatible ★ Uses CPM
★ Jumper Selectable ★ Persci, Shugart, etc.
Assembled and tested $269.95
Kit $179.95
Bare PC Board .... $ 39.95
NOTE: For CPM Add $70.00, Documentation Add $20.00
SD Versafloppy Kit $159.95
SD Versafloppy Assembled $189.95
★ Cassette I/O Kit only $115.00
★ 1771-01 Floppy Chip only .... $ 27.95

EXPANDORAM MEMORY KITS

★ Bank Selectable ★ Uses 4115 or 4116 200 ns.
★ Write Protect ★ Power 8VDC, ±16VDC
★ Phantom ★ Lowest Cost/Bit

Expando 32 Kit (4115)		Expando 64 Kit (4116)	
8K	$189.95	16K	$275.95
16K	$249.95	32K	$429.95
24K	$325.00	48K	$599.95
32K	$399.95	64K	$719.95

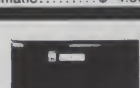
TRS 80 16K-UPGRADE KIT

★ 16K with Jumpers & Instructions for either Level I or Level II $89.95
★ 16K for Apple II Upgrade $89.95
Special: TRS80 Schematic .... $ 4.95

SHUGART DISK DRIVES

SA 400 5¼" Single Density $295.00
SA 801R 8" Single-Sided* $495.00
*Dual Drive Add on Mainframe inc. box, 2 drives, power, cable, fan $1295.00
Persci Model 277 Dual $1230.00
Persei Box & Power Supply .... $ 325.00

PET TO S-100 ADAPTER



Allows Pet to be interfaced to popular S-100 Bus.
Kit $189.95
Assembled $269.95
For Low Cost 8 Slot Bus to expand your Pet only .... $149.95 Kit

DC HAYES DATA COMMUNICATIONS ADAPTER

★ Telephone/TWX ★ S-100 compatible
★ Bell 103 freq. ★ Originate & answer mode
Assembled & Tested $279.95
AJ A242A Acoustic Coupler $325.00

SUPPORT DEVICES

AM9511 Arith Processor	$195.00
AM9517 DMA Controller	71.95
AM9519 Universal Interrupt	24.95
3881 (Z80 PIO)	10.45
3881-4 (4MHz)	14.95
3882 (Z-80 CTC)	10.45
3882-4 (4MHz)	14.95
8205/74S138 Decoder	2.95
8212 8 bit I/O	2.50
8214 Priority Int	4.95
8216 Bus Driver	2.50
8224 Clock Gen	2.95
8224-4 (4MHz)	9.75
8226 Bus Driver	3.95
8T26 Bus Driver	2.39
8228 Sys. Control	5.95
8238 Sys. Cont.	6.25
8251 Prog. I/O	6.95
8253 Int. Timer	19.50
8255 Prog. I/O	6.50
8257 Prog. DMA	19.50
8259 Prog. Int	19.50
8275 CRT Controller	74.95
8279 Prog. Keyboard	19.95
6810-1 128 x 8 RAM	4.75
6820 PIA	6.50
6821 PIA	7.25
6828 Priority Int	11.95
6834-1 512 x 8 Eprom	12.95
6850 ACIA	7.20
6852 Serial Adapter	9.95
6845/HD46505 CRT Contr	39.95
6860 Modem	9.95
6882 Modulator	11.95
6871A 1.0MHz OSC	25.95
6875	8.25
6880 Bus Driver	2.39
1821 SCD 1K RAM	25.00
1822 SCD 256 x 4 RAM	16.95
1824 CD 32 x 8 RAM	9.95
1852 CD 8 bit I/O	10.95
1856 CD I/O	8.95
1857 CD I/O	8.95
6520 PIA	7.50
6522 Mult.	9.25
6530-002	15.50
6530-003	15.50
6530-004	15.50
8530-005	15.50
3851 F8 Prog. Store	14.95
3853 F8 Memory I/O	14.95

RAMS DYN.

4116 16K x 1	14.95
4115 8K x 1	6.95
4050 4K x 1	4.45
4060 4K x 1	4.45
4096 4K x 1	4.25
2104 4K x 1	4.95
2107B-4 4K x 1	4.25
5261	1.95
5262	1.95
5270	4.95
5280	4.95
1103	1.95
4008L	4.95
TMS4070 16K x 1	19.95
4027-3	4.95
MCM6605 4K x 1	7.95
6002	1.50

CHARACTER GEN

RO32513-001 (5V) Upper	9.50
RO32513-005 (5V) Lower	10.95
RO32513-ADM3 (5V) Lower	14.95
MCM6571	10.75
MCM6571A	10.75
MCM6574	14.50
MCM6575	14.50

UARTS/USRTS

TR1602B (5V, 12V)	3.95
AY51013 (5V, 12V)	4.95
AY51014A/1612 (5-14V)	6.95
AY51015A/1863 (5V)	6.95
TMS 6011 (5V, 12V)	5.50
IM6402	9.95
IM6403	9.95
2350 USRT	9.95
1671B Astros	24.95
SALE TR1472B	**9.95**

ATTENTION TRS-80 USERS

Now in stock the hard to find 20/40 Memory Expansion Connector with contacts 7.95
★ TRS-80 Electric Pencil 99.95
★ Minifloppy-You provide expansion I/O and will install FD200 .... 425.00
★ TRS-80 16K Upgrade Kit 89.95
★ 16K Upgrade installed kit inc. labor 169.95
★ Keyboard and Video Mod to add lower case & eliminate horiz. smear 59.95
★ Clock Modification-30% faster 49.95
★ RS232 Printer I/O Modification w/software selectable baud rates up to 9600 .... 79.95
★ Level II mod. allows you to keep Level I and use switch to Level II 69.95

TERMS: Ship your TRS-80 to us insured and properly packaged. We will return freight collect. All warranty seals must be intact. Otherwise they will be returned. Allow 2 weeks turnaround

BAUD RATE GEN

MC14411	9.95
4702	14.95
WD1941	9.95

TV CHIPS

MM5320/21 TV Synch.	7.95
MM5369 Prescaler	3.95
LM1889 Modulator	3.95
CW300 Saw Function	1.50
MM57100 Color TV Game	8.95
MM57104 Clock	3.75
AY38500-1 TV Game	5.95
AY38600 Color TV Game	24.95
AY38615-1 Color Converter	8.95
AY38700-1/8710 Tank Chip	29.95
RF Modulator with Audio	8.95

A/D CONVERTERS

8700 8 bit Binary	13.50
8701 10 bit Binary	22.00
8703 8 bit TS	13.50
9400 Volt to Freq. Conv.	7.25
8750 3-1/2 Digit BCD	13.95
8038 Multifunction	3.95
1408L6 6 bit	3.95
1408L8 8 bit	5.95
DACOI A/D	5.95

TV INTERFACES

Pixie-Verter	8.50
TV-1 Video Interface	8.95
Pickles & Trout TV Mod. Kit	20.00
Microverter	35.00
M&R Modulator	35.00

MICROPROCESSORS

Z-80	**$19.95**
Z-80A	24.95
F-8 (3850)	16.95
2650	18.95
CD1802	19.95
8080A	9.95
8080A-4MHz	19.95
SALE 8085	**18.95**
8008-1	14.95
2901	16.95
2901A	24.95
TMS 9900JL	69.95
CP1600	39.95
6502	11.50
8502A	19.95
IM6100	29.95
6800	16.95
6802P	24.95
8035	17.50
8755	49.95
8748	89.95

SOCKETS

8 Pin W/W	32	8 Pin S/T	17
14 Pin W/W	.37	14 Pin S/T	20
16 Pin W/W	38	16 Pin S/T	22
18 Pin W/W	60	18 Pin S/T	31
20 Pin W/W	90	20 Pin S/T	34
22 Pin W/W	93	22 Pin S/T	35
24 Pin W/W	85	24 Pin S/T	41
28 Pin W/W	1.15	28 Pin S/T	49
40 Pin W/W	1.49	40 Pin S/T	63

CONNECTORS

6 Pin Single S/E	1.49
15/30 Dual S/E	1.95
18/36 Dual S/E	2.35
22/44 Dual S/E	2.95
40/80 W/W	4.95
43/86 Dual W/W/S/T	8.50
50/100 IMSAI W/W	4.75
50/100 IMSAI S/T	4.75
50/100 Altair W/W/S/T	5.95
IMSAI Card Guides	4/1.00

25 Pin D Subminature	
DB25P	3.25
DB25S	3.75
Hood	1.25
Set w/Hood	6.50

CTS DIPSWITCHES

CTS206-4 $1.75 | CTS206-8 $1.95
CTS206-5 $1.75 | CTS206-9 $1.95
CTS206-6 $1.75 | CTS206-10 $1.95
CTS206-7 $1.75

LIVERMORE BASIC

ON 65K ROM
THAT'S RIGHT ON ROM!!!
OUR PRICE only $95.00

KEYBOARD ENCODERS

AY5-2378	13.75
AY5-3600	13.75

NAKED PC BOARD SALE

Z-80 CPU (Ithaca)	$34.95
8080A CPU	34.95
8K Static RAM (Logos)	21.95
16K Static RAM (2114)	29.95
32K Static RAM (2114)	49.95
Floppy I/O (Tarbell)	39.95
Cassette I/O (Tarbell)	29.95
8K Eprom (2708)	21.95
1702 Eprom Board	30.00
2708/2718 Eprom (Ithaca)	34.95
2708/2718 Eprom (WMC)	30.00
Realtime Clock	34.95
ACP Proto Bd. (3M Conn.)	27.95
Vector 8800 Proto.	19.95
Vector 8803 11 slot MB	29.95
ACP Extender w/Conn.	15.95
Video Interface (SSM)	27.95
Parallel Interface (SSM)	27.95
13 Slot MotherBoard (WMC)	32.95
9 Slot MotherBoard (WMC)	29.95
8 Slot Mother (expandable)	34.95

ATTENTION KIM USERS

KIMSAI-expansion to S-100	125.00 Kit/165.00
KIMSI to KIM Connector	5.75/pair
KIM 1 6502 Single Board Computer	179.00
KIM 1 Power Supply	59.95
KIM Memory Plus - (consists of 8K Ram 8K 2716 Eprom, Programmer I/O etc.)	245.00
KIM SOFTWARE	
• Please package (cassette) 12 games	16.95
• Help Editor package (cassette)	16.95
• Help Mailing List pkg. (cassette)	16.95
• Help Info Retrival pkg. (cassette)	16.95
• Microchess (cassette)	16.95
• Microsoid Assembly/Disassem/Editor	27.95
• Microsoid Source Listing (cassette)	27.95
• Tiny Basic for KIM (paper tape)	10.95

FLOPPY DISK I/O

1771-01 8" & Minifloppy	27.95
uPd372 Nec Floppy	49.95
1781 Dual Floppy	39.95
1791 Dual Floppy	99.00

ATTENTION PET USERS

BETSI-pet expansion to S100	105.00 Kit/180.00
PET Connector Kit includes (4) Connectors for memory expansion, IEEE 488 I/O cassette I/O and parallel user port	7.95 /set
Video Buffer (converts to Std. Video)	29.95
Petunia (Music Board)	29.95
Combo (Video & Petunia)	49.95
Beeper (signals tape load)	24.95

ATTENTION APPLE II USERS

Apple II w/16K	$1045.00
16K Upgrade Kit	89.95
Floppy Disk II w/Interface	595.00
Floppy Disk II	495.00
Firmware Card	180.00
Video Monitor (B/W)	205.00
Printer Centronics P-1 w/cont	895.00
Printer Centronics P-1	475.00
Printer Centr. 779 w/cont.	1445.00
Apple II Proto Board	24.00
Parallel I/O Card	180.00
Communications I/O Card	180.00
PR40 Printer I/O w/cassette	49.95
2716 Eprom Burner w/adapt	99.95
Apple II to MFE tape I/O	197.50
Coresident Assembler Cass	19.95
Heuristics Robot Car	249.00
Heuristics 20A Speechlab	189.00
Dow Jones Stock Quote Pkg	25.00
AJ A242 Coupler	325.00
I/O to transmit over existing wiring	189.00 A/T

All Shipments FCM or UPS. Orders under $100.00 add 5% handling and postage. Orders over $100.00 add 2.5% handling & postage. Mastercharge/Bankamericard/COD accepted w/25% deposit. California Residents add 6% tax. Foreign Orders add 8% handling. All parts prime factory tested guaranteed. Same day shipment. Add 25 cents for Data.
Retail pricing may vary from Mail Order Pricing. All pricing subject to change without notice.

STATIC RAM HEADQUARTERS

	1-24	25-99	100
21L02 450ns.	1.30	1.25	1.15
21L02 250ns	1.59	1.55	1.45
2102	1.25	1.15	1.10
2111	3.75	3.65	3.55
2112-1	2.95	2.85	2.85
2101-1	2.90	2.70	2.55
2114-250ns.	9.95	7.95	7.25
2114-450ns.	7.95	6.95	5.25
TMS 4044/5257 250ns.	9.95	8.95	7.95
TMS 4044/5257 450ns.	7.95	7.25	5.95
EMM4200A	9.75	8.75	7.95
EMM4402	7.95	7.25	6.25
EMM4804	12.50	11.50	9.95
5101C-E	7.95	7.95	7.25
upd410 (4200)	10.95	10.25	9.25
AMD9140/41	10.95	10.25	9.25
AMD9130/31	12.95	11.95	10.25
FSC 460/46416K CCD Only $18.95 Each			
1101	1.95	1.75	1.25
P2125/93425 (45 ns)	7.95	7.35	7.25
8508 1K x 1 CMOS	7.95	7.95	7.25
6518 1K x 1 CMOS	7.95	7.95	7.25
74S189 64 bit Ram	3.95	3.25	2.50
8155 I/O w/Ram 21.95			

WAVEFORM GENERATORS

8038 Function Gen.	3.95
MC4024 VCO	2.45
LM568 VCO	1.75
XR2208 Function Generator	5.25

CHARGE COUPLED DEVICES

16K CCD - First time offered Fairchild 460 CCD 16K Memory (now you can experiment with CCD technology at a reasonable price. 17 page Application note supplied with each order. Quantity limited)

$18.95 each (reg. 43.00)

CRYSTALS

Microprocessor Timebases TV Game

Frequency	Price	Frequency	Price
1.0MHz	$5.85	6.0MHz	$ 4.95
1.8432	4.95	6.144	4.95
2.0MHz	5.85	6.5536	4.95
2.01MHz	2.95	10.0MHz	4.95
2.097152MHz	5.85	13.0MHz	4.95
2.4576MHz	5.85	14.31818	4.95
3.579545MHz	1.50	18.0MHz	4.95
4.0MHz	4.95	18.432MHz	5.95
4.194304MHz	5.95	20.0MHz	4.95
4.91520MHz	5.95	22.1184MHz	5.95
5.0MHz	4.95	27.0MHz	5.95
5.0688	4.95	36.0MHz	5.95
5.7143MHz	5.95	48.0MHz	5.95
		100KC	12.95

DISPLAYS/OPTO

DL 704/707/CC/CA 300	1.25
FND359 CC .357	.95
FND 500/507/CC/CA 500	1.35
FND 503/510 CC/CA 500	.95
FND 800/807 CC/CA 800	2.50
Bowmar 9 digit bubble	.99
FSC 8024 4 digit CC 800	4.95
HP7340 HEX Display	19.95
HP 5082-7340 HEX	19.95
HP 5082-7300 HEX	19.95
TIL 305 5x7 Array	5.25
TIL 308, 308, 309.7 seg.	8.95
TIL 311 HEX display	12.95
MA 1003 12 auto clock	17.95
MA 1002, 1010 4 digit clock module	9.95
Bezel for above	4.95
NSN 373/374 dual CC/CA 300	2.20
NSN 583/584 dual CC/CA 500	2.60
NSN 783/784 dual CC/CA 700	3.00
4N25 Opto Isolater	1.10
MCT 2 Opto Isolater	.89
4N33 Darlington ISOL	1.75
Red Led's 185 Dia	5/1.00
Green/Yellow	4/1.00
HP 5082/7731 7 seg	.90
MAN IOA	9.95

MONTHLY SPECIALS

22 Pin S/T Sockets	.10
1488 or 1489 RS232	.95
1408L8	4/19.95
8T26	2.25
ICL 7208	13.95
ICL 7107 3½ Dig. A/D	11.95
ICL 8211 Volt. Ref.	1.95
LM1877 Dual 2W Amp.	1.95
LM390 Batt. Op. Audio Amp	1.75
LM1830 Fluid Detector	1.95
LM1850 Ground Fault IC	1.75
LM1800 PLL FM Stereo	2.25
LM1820 AM Radio	1.25
LM2917 Freq to Voltage	2.25
MK5014 Calculator IC	.95
75451/52/53	10/2.50
8T20 Bidirector one shot	3.49
MH0025 or MH0026 Mos Driver	2.50
8223 Prom	2.95

NEW
1978 CATALOGUE
SEND 25¢ POSTAGE

DISCOUNT COMPUTER CORNER

	Discount Cash	Computer Credit Card
Apple II/16K	$1045.00	$1065.00
Compucolor II	Contact sales desk	
Exidy	Contact sales desk	
Soroc IQ 140	1495.00	1495.00
TEI PT208	4599.00	4699.00
TEI MCS122	450.00	462.00
Cromemco Z-2	536.00	551.00
Cromemco SYS III	5499.00	5561.00
Centronics P-1	475.00	487.00
Centronics S-1	525.00	539.00
TI 810 Printer	1895.00	1925.00
Persci 277	1230.00	1266.00
Shugart 800/801	495.00	512.00
Pertec FD200	299.00	319.00
MFE Dual 751	730.00	744.95
Horizon I	1439.00	1479.00
North Star	599.00	617.00
ADM 3 Assem	829.95	865.00
Soroc IQ 120	895.00	945.00
Teletype Model 43	1199.00	1230.00
AM100	Contact sales desk	
Imsai w/22 slots	599.00	617.00
Oliver 2708/16 Programr	275.00	288.00
Houston Digital Plotter	999.00	999.00
Pennywhistle Modem	139.95	139.95
Hazeltine 1500	1175.00	1195.00
Okidata 110 w/TF RS232	1675.00	1705.00
Okidata 22 w/TF RS232	2705.00	2810.00
MEK 6800 D2 Kit	235.00	Same
KIM I 6502	199.00	Same
Intercept Jr 6100	281.00	Same
Technico 9900 Kit	299.00	Same
RCA VIP Cosmac	249.00	Same
Rockwell AIM	375.00	Same
Synertek VIM-1	249.00	Same

DATABOOKS & MANUALS

NSC TTL Data	$ 3.95	AMI MOS/LSI Data	3.95	Mot. Vol 4 Mecl Data	3.95
NSC Linear Data	4.95	GI MOS/LSI Data	4.95	Mot. Vol. 5 CMOS Data	2.95
NSC Linear APP/note I	3.95	Osborne Intro to Micro Vol. 0	8.50	Mot. Vol. 8 Linear Data	3.95
NSC Linear APP/note II	3.95	Osborne Intro to Micro Vol. I	8.50	Mot. Vol. 9 Schottky TTL	2.95
NSC CMOS Data	2.95	Osborne Intro to Micro Vol. II	15.00	Mot. MPU Applications	25.00
NSC Audio Data	3.95	Osborne 8080 Programming	8.50	Mot. MPU Prog. Ref. Manual	3.95
NSC Volt. Reg. Data	2.95	Osborne 6800 Programming	8.50	Mot. Power Data	2.95
NSC Memory Data	3.95	Osborne Z80 Programming	8.50	Mot. Rectifier Data	2.95
NSC MOS/LSI Data	3.95	TI Powr Semi's Data	7.50	Mot. Switching Tran	2.95
NSC Power Transistors	2.95	TI TTL Data	6.95	Mot Zeners	2.95
Intel Databook	3.95	TI Transistors & Diodes	8.50	Basic Software SRI Vol. I or II	24.95
Intel MCS85 Manual	7.50	TI Memory Data	3.95	Basic Software SRI Vol. III	39.95
Intel MCS80 Manual	7.95	TI Optoelectronics	3.95	Basic Software SRI Vol. IV or V	9.95
Intel MCS40 Manual	4.95	TI Linear Data	3.95	Basic Software SRI Vol. VI	49.95
AMD 8080 Handbook	5.95	TI Bipolar Memory	3.95	Basic Software SRI Vol. VII	39.95
AMD Linear Data	4.95	TI Interface Data	4.95	1978 IC Master	47.50
AMD Schottky Data	4.95	Motorola Semi Data 1 2 3	9.95		

DISKETTES

5" MINI
★ Soft Sector
★ 10 Sector
★ 16 Sector
$4.50 each $3.90/10

NORTH STAR MINI
$4.95 each $4.50/10

8" STANDARD
★ Hard
★ Soft
$4.50 each $4.10/10

6800 DESIGNER BOARDS MODULES PROTO BOARDS

★ Motorola Compatible Modules*

MEK 6800 D2 Kit	$235.00
9600 6800 MPU Module	495.00
9601 16 slot Mother Bd	175.00
9602 16 slot Card Cage	75.00
9603 8 slot Mother Bd	99.00
9604 System Power Supply	250.00
9610 Prototype Board	36.00
9615 4K Eprom Module	250.00
9620 16 port parallel I/O	375.00
9626 8K Static RAM	295.00
9626K 8K Static RAM Kit	225.00
9630 Extender Card	60.00
9640 Multiple Tuner Prog	395.00
9650 8 port Duplex Asyn I/O	395.00
Mot 43/86 Connectors w/wor s/t	5.95
AMI EVK 99* 6800 sub Kit	119.00
AMI EVK 300 Assembled	495.00
EVK Connectors	6.50
AMI 6800 Proto Rom	30.00
AMI 6800 Micro Assembler Rom	30.00
6800 Tiny Basic Paper Tape	20.00
6800 Tiny Basic Eprom	125.00
EVK Accessory Boards consult sales desk	

S-100 MOTHERBOARD SPECIAL

8 slot expandable w/9 conn.
reg $69.95, NOW $52.95

PCG PROGRAMMABLE CHARACTER GENERATOR

The hottest version of STARWARS available from Objective Design.
S-100 Compatible 2MHz Kit with object code on tarbell or cuts $169.95

UV "Eprom" Eraser

Model UVs-11E **$64.95**
Holds 4 Eprom's at a time
Backed by 45 years UV experience
Model S-52T **$219.95**
Professional Industrial Model

PROMS

2708	9.95
2708-6	5.95
1702A	3.95
1702A-6	3.50
2716-5V	44.95
2716-5V, 12V	24.95
2758 5V	25.95
5203AQ	11.95
5204AQ	9.95
6834-1	12.95
IM 5610	2.95
SALE 8223	**2.95**
82S123	2.95
82S126	4.50
82S129	4.95

P. O. BOX 17329 Irvine, California 92713 DEPT K Phone (714) 558-8813
TWX: 910-595-1565

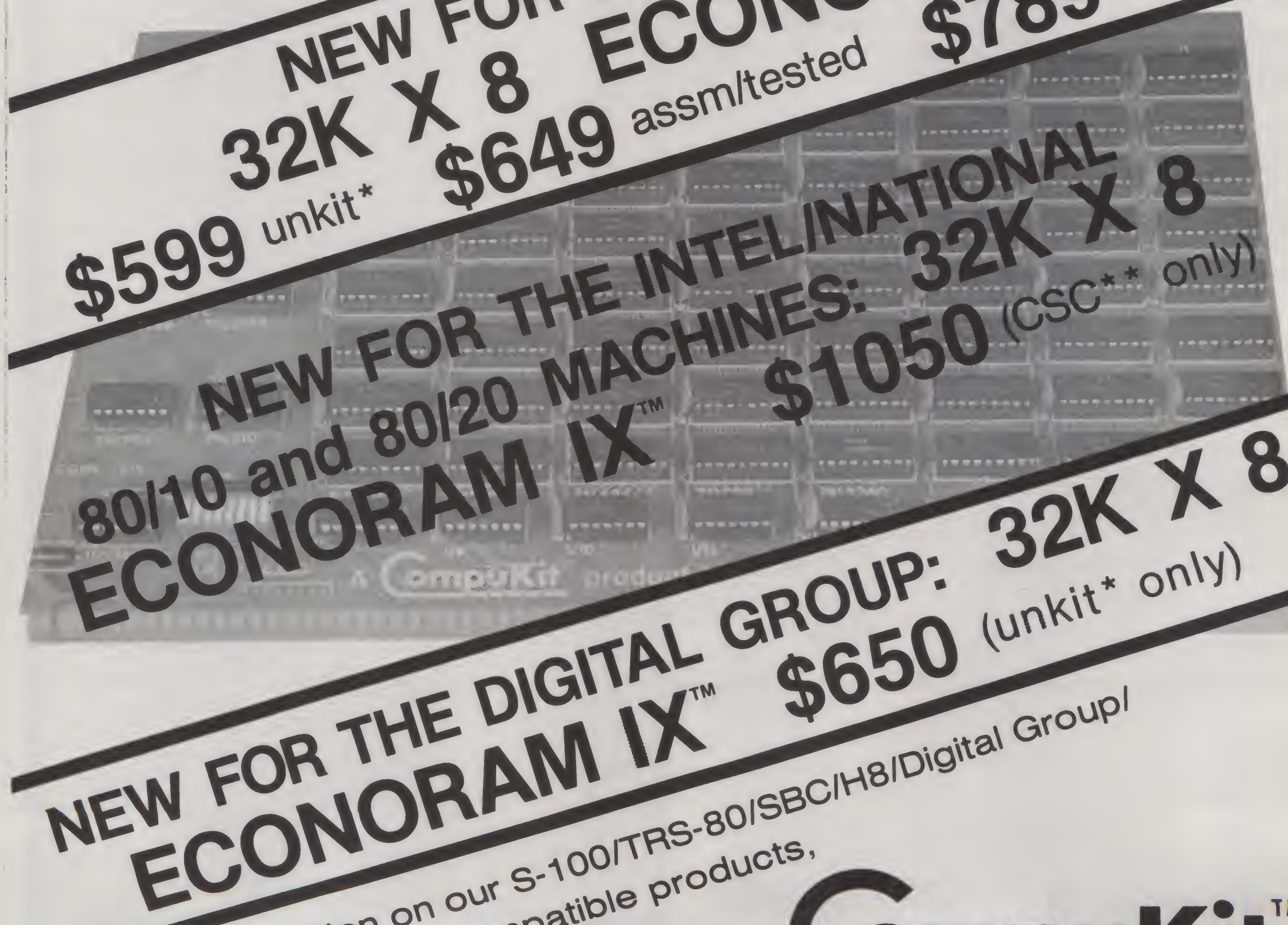
Retail Store Open Mon. – Sat.
Located at 1310 "B" E. Edinger,
Santa Ana, CA 92705

CompuKit™ DELIVERS

32K STATIC MEMORIES

We're best known for reliability and quality . . . now you can add density to the list. Not just for one buss, either, but for **three** different busses. And, all these oversized memories fit on regular sized cards: you don't have to take the cover off your machine when you run a CompuKit 32K board.

Our memories are fully static, buffered, run at 4 MHz, feature low power consumption, and use only the best mechanical and electronic components. These boards give you the density, variety, and performance.



NEW FOR THE S-100 BUSS: 32K X 8 ECONORAM X™

$599 unkit* $649 assm/tested $789 CSC**

NEW FOR THE INTEL/NATIONAL 80/10 and 80/20 MACHINES: 32K X 8 ECONORAM IX™

$1050 (CSC** only)

NEW FOR THE DIGITAL GROUP: 32K X 8 ECONORAM IX™

$650 (unkit* only)

For more information on our S-100/TRS-80/SBC/H8/Digital Group/ Apple compatible products,

call or write today.

CompuKit™

Box 2355, Oakland Airport, CA 94614

from

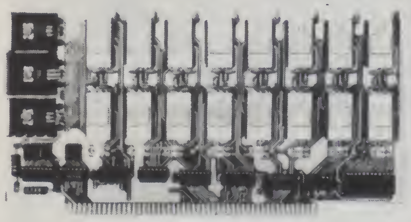
GODBOUT

G4

***Unkit:** Sockets, bypass capacitors pre-soldered in place for easy one evening assembly.

****CSC:** Qualified under the Certified System Component program. These boards are burned in for 200 hours minimum, serial numbered, guaranteed to run at 4 MHz over the full commercial temperature range, and are immediately replaced if failure occurs within 1 year of invoice date.

16K EPROM CARD-S 100 BUSS



$59.95 KIT

OUR BEST SELLING KIT!

USES 2708's!

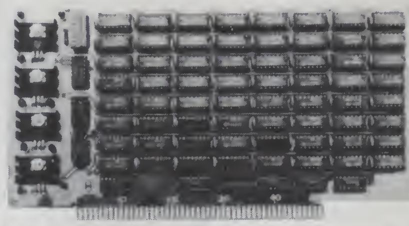
Thousands of personal and business systems around the world use this board with complete satisfaction. Puts 16K of software on line at **ALL TIMES!** Kit features a top quality soldermasked and silk-screened PC board and first run parts and sockets. All parts (except 2708's) are included. Any number of EPROM locations may be disabled to avoid any memory conflicts. Fully buffered and has WAIT STATE capabilities.

OUR 450NS 2708'S ARE $8.95 EA. WITH PURCHASE OF KIT

ASSEMBLED AND FULLY TESTED ADD $25

8K LOW POWER RAM KIT-S 100 BUSS

250 NS SALE!



ADD $5 FOR 250NS!

$129 KIT

Use 21L02 450 NS RAMS!

Thousands of computer systems rely on this rugged, work horse, RAM board. Designed for error-free, NO HASSLE, systems use.

KIT FEATURES:

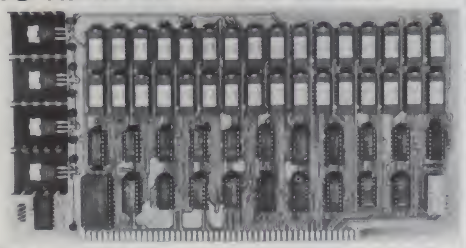
1. Doubled sided PC Board with solder mask and silk screen layout. Gold plated contact fingers.
2. All sockets included.
3. Fully buffered on all address and data lines.
4. Phantom is jumper selectable to pin 67.
5. FOUR 7805 regulators are provided on card.

Blank PC Board w/Documentation **$29.95**
Low Profile Socket Set...**13.50**
Support IC's (TTL & Regulators) **$9.75**
Bypass CAP's (Disc & Tantalums) **$4.50**

ASSEMBLED AND FULLY BURNED IN ADD $30

16K STATIC RAM KIT-S 100 BUSS

$295 KIT



FULLY STATIC, AT DYNAMIC PRICES

WHY THE 2114 RAM CHIP?
We feel the 2114 will be the next industry standard RAM chip (like the 2102 was). This means price, availability, and quality will all be good! Next, the 2114 is FULLY STATIC! We feel this is the **ONLY** way to go on the S-100 Buss! We've all heard the HORROR stories about some Dynamic Ram Boards having trouble with DMA and FLOPPY DISC DRIVES. Who needs these kinds of problems? And finally, even among other 4K Static RAM's the 2114 stands out! Not all 4K static Rams are created equal! Some of the other 4K's have **clocked** chip enable lines and various timing windows **just as critical** as Dynamic RAM's. Some of our competitor's 16K boards use these "tricky" devices. But not us! The 2114 is the ONLY logical choice for a trouble-free, straightforward design.

KIT FEATURES:
1. Addressable as four separate 4K Blocks.
2. ON BOARD BANK SELECT circuitry. (Cromemco Standard!). Allows up to 512K on line!
3. Uses 2114 (450NS) 4K Static Rams.
4. ON BOARD SELECTABLE WAIT STATES.
5. Double sided PC Board, with solder mask and silk screened layout. Gold plated contact fingers.
6. All address and data lines fully buffered.
7. Kit includes ALL parts and sockets.
8. PHANTOM is jumpered to PIN 67.
9. LOW POWER: under 2 amps TYPICAL from the +8 Volt Buss.
10. Blank PC Board can be populated as any multiple of 4K.

BLANK PC BOARD W/DATA—**$33**
LOW PROFILE SOCKET SET—**$12**
SUPPORT IC'S & CAPS—**$19.95**
ASSEMBLED & TESTED—**ADD $30**
2114 RAM'S—**8 FOR $69.95**

16K STATIC RAM KIT SWTPC (SS-50) 6800 BUSS

USES 2114 4K RAMS!

$295 COMPLETE KIT

WHY PAY MORE FOR FINNICKY DYNAMIC BOARDS?

At last an affordable static RAM board for this popular buss. Quality PC Board with solder mask and silk-screen. Fully buffered with plenty of bypassing for reliable operation. FOUR ON-BOARD REGULATORS.

16K DYNAMIC RAM CHIP

16K X 1 Bits 16 Pin Package. Same as Mostek 4116-4 250 NS access. 410 NS cycle time. Our best price yet for this state of the art RAM 32K and 64K RAM boards using this chip are readily available. These are new, fully guaranteed devices by a major mfg. **VERY LIMITED STOCK!**

8 FOR $89.95

NOT ASSOCIATED WITH DIGITAL RESEARCH OF CALIFORNIA, THE SUPPLIERS OF CPM SOFTWARE.

2708 EPROMS

450 NS!

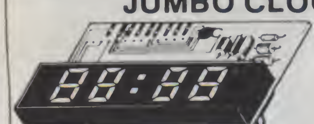
Now full speed! Prime new units from a major U.S. Mfg. 450 N.S. Access time. 1K x 8. Equiv. to 4-1702 A's in one package.

~~$15.75 ea.~~ **$9.95** ~~4 FOR $50.00~~

PRICE CUT

NATIONAL SEMICONDUCTOR JUMBO CLOCK MODULE

MA1008A BRAND NEW!



$6.95

2 FOR $13

(AC XFMR $1.95)

ASSEMBLED! NOT A KIT!

PERFECT FOR USE WITH A TIMEBASE.

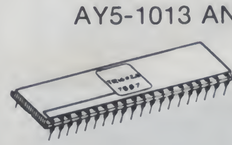
COMPARE AT UP TO TWICE OUR PRICE!

MANUFACTURER'S CLOSEOUT!

#MA1008D — $9.95

WESTERN DIGITAL UART

TR1602A. PIN FOR PIN SUB FOR AY5-1013 AND TMS6011.
FOR SERIAL I/O



$2.99 EACH

SURPLUS SPECIAL

SALE!
1N4148 DIODES. SILICON. Same as 1N914. New, factory prime, Full Leads.
100 FOR $2
1000 FOR $17.50

New! REAL TIME Computer Clock Chip
N.S. MM5313. Features BOTH 7 segment and BCD outputs. 28 Pin DIP. **$4.95 with Data**

Z-80 PROGRAMMING MANUAL

By MOSTEK, or ZILOG. The most detailed explanation ever on the working of the Z-80 CPU CHIPS. At least one full page on each of the 158 Z-80 instructions. A MUST reference manual for any user of the Z-80. 300 pages. Just off the press. **$12.95**

COMPUTER PARTS

Z-80	**19.95**	8212	**2.25**
Z-80A	**24.95**	8255	**6.95**
8080A	**6.95**	2111AL-4	**2.25**
8080A-2	**8.95**	2708	**9.95**

MOTOROLA 7805R VOLTAGE REGULATOR. Same as standard 7805 except 750 MA output. TO-220. 5VDC output.

44¢ each or 10 for $3.95

Digital Research Corporation

(OF TEXAS)

P.O. BOX 401247K GARLAND, TEXAS 75040 • (214) 271-2461

D20

TERMS: Add 30¢ postage, we pay balance Orders under $15 add 75¢ handling No C.O.D. We accept Visa, MasterCharge, and American Express cards Tex Res add 5% Tax Foreign orders (except Canada add 20% P & H 90 Day Money Back Guarantee on all items

P.O. Box 4430M Santa Clara, CA 95054
For will call only: (408) 988-1640
2322 Walsh Ave.

Quest
ELECTRONICS

Same day shipment. First line parts only. Factory tested. Guaranteed money back. Quality IC's and other components at factory prices. Q3

INTEGRATED CIRCUITS

7400TTL

7400N	.17
7402N	.17
7404N	.19
7409N	.23
7410N	.17
7414N	.63
7420N	.17
7422N	1.39
7430N	.20
7442N	.50
7445N	.89
7447N	.60
7448N	.89
7450N	.17
7474N	.29
7475N	.49
7485N	.88
7489N	2.00
7490N	.43
7492N	.43
7493N	.43
7495N	.69
74100N	.90
74107N	.29
74121N	.34
74123N	.59
74125N	.39
74145N	.89
74150N	.95
74151N	.69
74154N	1.00
74157N	.89
74161N	.87
74162N	.87
74183N	.87
74174N	.96
74175N	.90
74190N	1.15
74192N	.87
74193N	.85
74221N	1.55
74298N	1.85
74365N	.66
74366N	.66
74367N	.66

74LS00 TTL

74LS00N	.25
74LS02N	.25
74LS04N	.25
74LS05N	.25
74LS08N	.25
74LS10N	.25
74LS13N	.40
74LS14N	.90
74LS20N	.25
74LS22N	.25
74LS28N	.41
74LS30N	.25
74LS33N	.39
74LS38N	.30
74LS74N	.35
74LS75N	.47
74LS90N	.51
74LS93N	.51
74LS95N	1.89
74LS107N	.35
74LS112N	.35
74LS113N	.35
74LS132N	.72
74LS136N	.35
74LS151N	.87
74LS155N	.87
74LS157N	.87
74LS182N	.91
74LS183N	.91
74LS174N	.95
74LS190N	1.06
74LS221N	1.95
74LS258N	.87
74LS367N	.89

LINEAR

CA3045	.90
CA3048	.67
CA3081	1.80
CA3082	1.90
CA3089	2.95
LM301AN/AH	.35
LM305H	.87
LM307N	.35
LM308N	.69
LM309H	1.15
LM309K	.95
LM311H/N	.90
LM317T/K	2.92
LM318	1.35
LM320K-5	1.20
LM323K-5	8.95
LM320K-12	1.35
LM320K-15	1.35
LM320T-5	1.60
LM320T-8	1.60
LM320T-12	1.50
LM320T-15	1.60
LM324N	1.15
LM339N	1.55
LM340K-5	1.10
LM340K-8	1.10
LM340K-12	1.10
LM340K-15	1.10
LM340K-24	1.10
LM340T-5	1.10
LM340T-8	1.10
LM340T-12	1.10
LM340T-15	1.10
LM340T-18	1.10
LM340T-24	1.10
LM343H	4 50
LM370	1.15
LM377	4 50
LM379	5.00
LM380N	1.00
LM381	1.60
LM382	1.60
LM703H	40
LM709H	.28
LM723H/N	50
LM733N	.87
LM741CH	.35
LM741N	.25
LM747H/N	82
LM748N	.35
LM1303N	.82
LM1304	1.10
LM1305	1 27
LM1307	2.00
LM1310	2.75
LM1458	.47
LM1800	1.75
LM1812	7.50
LM1889	3.00
LM2111	1.75
LM2902	1.50
LM3900N	60
LM3905	1.75
LM3909N	81
MC1458V	.50
NE540L	2.89
NE550N	85
NE555V	43
NE556A	.79
NE565A	1.00
NE566V	1.50
NE567V	1.20
NE570B	5.00
NE571B	5.00
78L05	60
78L08	60
79L05	.70
78M05	.85
75108	1.75
75491CN	.50
75492CN	.55
75494CN	89

A to D CONVERTER

8038B	4.50
8700CJ	13.95
8701CN	22.00
8750CJ	13.95
LD130	9.95
9400CJV/F	7.40
ICL7103	9.50
ICL7107	14.25

CMOS

CD34001 Fair.	.50
CD4000	.18
CD4001	.21
CD4002	.21
CD4006	1.10
CD4007	.21
CD4008	.21
CD4009	.39
CD4010	.39
CD4011	21
CD4012	.21
CD4013	.38
CD4014	.86
CD4015	.88
CD4018	.36
CD4017	.94
CD4018	.94
CD4019	.21
CD4020	1.02
CD4021	1.02
CD4022	.86
CD4023	.21
CD4024	.75
CD4025	.21
CD4028	1.51
CD4027	.36
CD4028	.79
CD4029	1.02
CD4030	.21
CD4035	1.02
CD4040	1.02
CD4042	.71
CD4043	.63
CD4044	.83
CD4048	1.87
CD4049	.36
CD4050	.38
CD4051	1.13
CD4060	1.42
CD4066	.71
CD4068	.40
CD4069	.40
CD4070	40
CD4071	.21
CD4072	.21
CD4073	.21
CD4075	.21
CD4078	1.75
CD4078	40
CD4081	.21
CD4082	.21
CD4118	.47
CD4490	5.50
CD4507	1.00
CD4508	4.25
CD4510	1.02
CD4511	.94
CD4515	2.52
CD4516	1.10
CD4518	1.02
CD4520	1.02
CD4527	1.51
CD4528	.79
CD4553	5.75
CD4566	2.25
CD4583	4.50
CD4585	1.10
CD40192	3.00
74C00	.28
74C04	.33
74C10	.28
74C14	2.10
74C20	28
74C30	.28
74C48	1.95
74C74	.75
74C76	1.40
74C90	1.15
74C93	1.40
74C154	3.00
74C160	1.44
74C175	1.35
74C192	1 65
74C221	2.00
74C905	3.00
74C906	.75
74C914	1.95
74C922	5.50
74C923	5 50
74C925	6 95
74C928	8 95
74C927	8 95

INTERFACE

8095	85
8096	85
8097	85
8098	.85
8T09	1.25
8T10	4 50
8T13	3 00
6T20	5.50
8T23	3.10
8T24	3.50
8T25	3 20
8T26	1.89
8T28	2 75
6T97	1.89
8T98	1 89

MOS/MEMORY RAM

2101-1	3.95
2102-1	.95
2102AL-4	1.80
21F02	1.25
2104A-4	4 95
2107B	4 95
2111-1	3.75
2112-2	3 95
2114	8 50
4118	17.95
2513B	8.30
21L02-1	1.49
MM5282	.40
MM5280	3.00
MM5320	9 95
MM5330	5 94
PD411D-3	4 00
PD411D-4	5.00
P5101L	13 95
4200A	9 95
82S25	2.90
91L02A	1.75
HD0185-5	8.95
MM57100	4 50
GIAY38500-1	9 95
MCM8571A	9.95
9368	3.50
410D	10.00
418	18 00

CLOCKS

MM5309	3.00
MM5311	3.80
MM5312	4 80
MM5313	3.60
MM5314	3.90
MM5315	4 00
MM5316	5.00
MM5318	3.60
MM5369	2.10
MM5841	14 45
MM5865	7.95
CT7001	5 80
CT7002	8.95
CT7010	8 95
CT7015	7.25
MM5375AA/N	3 90
MM5375AB/N	4 90
7205	18.50
7207	7.50
7208	15.95
7209	4 95
DS0028CN	3.75
DS0056CN	3.75
MM53104	2.50

MICROPROCESSOR

6800	17.50
6802	24 95
8080A with data	8.95
8085	27.00
Z80	29.95

8212	2.90
8214	8.00
8216	2 90
8224	2 90
8228	5 35
8251	8 50
8253	10.00
8255	9 25
8257	19 50
8259	19 50
CDP1802CD	19.95
CDP1802D	25.00
CDP1861	12 95
6820	9.95
6850	12.95
6502	12.50

IC SOCKETS
Solder Tin Low Profile

PIN	1 UP	PIN	1UP
8	.15	22	.30
14	.18	24	35
16	20	28	.42
18	27	36	58
20	29	40	.57

3 level wire wrap gold
14 pin 25 16 pin 27
2 level 14 pin ww 20

WIRE WRAP LEVEL 3

PIN		PIN	
14	.25	24	86
16	.27	28	1.00
18	57	40	1 23

UART/FIFO

AY5-1013	5.50
AY5-1014	7 50
3341	8 95

PROM

1702A	3.95
N82S23	2 95
N82S123	3 50
N82S126	3 75
N82S129	3.75
N82S131	3 75
N82S136	8 75
N82S137	8.75
2708	12.50
DM8577	2.90
8223	2.90
2716T1	22.50
2716 Intel	48.00

CRYSTALS

1 MHz	4.50
2 MHz	4.50
4 MHz	4.25
5 MHz	4.25
10 MHz	4 25
18 MHz	3.90
20 MHz	3.90
32 MHz	3.90
32768 MHz	4 00
1.8432 MHz	4.50
3.5795 MHz	1.20
2.0100 MHz	1.95
2.097152 MHz	4.50
2 4576 MHz	4 50
3.2768 MHz	4.50
5.0688 MHz	4 50
5.185 MHz	4 50
5.7143 MHz	4.50
8.5536 MHz	4.50
14 31818 MHz	4.25
18 432 MHz	4.50
22.1184 MHz	4 50

CONNECTORS

44 pin edge	2.00
100 pin edge	4.50
100 pin edge WW	4 50

KEYBOARD ENCODERS

AY5-2376	$12.50
AY5-3600	13 50
74C922	5 50
74C923	5 50
HD0165-5	8 95

IC Test Clips

	1	10
Red	55	47
Black	.55	.47

Keyer 8043 14 50
comp. w/spec./sock.

TRANSISTORS

2N1893	40
2N2222A	18
2N2369	30
2N2904A	.20
2N2907A	.25
2N3053	40
2N3638	.25
2N3643	.25
2N3904	18
2N3906	18
2N3055	.89
2N4400	25
2N4401	75
2N4402	.20
TIP31	.60
TIP33A	1.00

CLOCK MODULES Complete alarm clocks ready to hook up with transformer and switches Very compact with .50" and 84" digits

MA1002A, C or E .50"	**8.95**
102P3 Transformer	**2.25**
MA1010A, C or E .84"	**11.95**
102P2 Transformer	**2.25**
Special transformer and six switches when purchased w/module	**2.95**
MA1003 car module .3" green fluor. display	**15.85**

RESISTORS ¼ watt 5%
10 per type .03 1000 per type .012
25 per type .025 350 piece pack
100 per type .015 5 per type 8.75

KEYBOARDS

56 key ASCII keyboard kit	$62.50
Fully assambled	72.50
53 key ASCII keyboard kit	55.00
Fully assembled 85.00 Enclosure	14.95

LEDS

Red T018	.15
Green, Orange, Yellow T018	.20
Jumbo Red	20
Green, Orange, Yellow Jumbo	.25
Clipite LED Mounting Clips	8/$1 25

(specify red, amber, green, yellow, clear)

CONTINENTAL SPECIALTIES in stock
Complete line of breadboard test equip
MAX-100 8 digit Freq. Ctr. $128.95

OK WIRE WRAP TOOLS in stock
Portable Multimeter $18.00

DIGITAL THERMOMETER $48.50
Batt. oper. General purpose or medical 32°–230°F. Disposable probe cover ±.2° accuracy. Comp. Assy in compact case

COMPUTER BOARD KITS

8K RAM Board Kit	$134 95
4K EPROM Kit	114 95
I/O Board Kit	44 50
Extender Board w/connector	12.50
Video Interface board kit	125.00
16K EPROM board kit w/o PROMS	74 50
18K Static RAM board kit	395 00
North Star Floppy Disk Kit	$665.00
Additional Drive Kit	415.00

SPECIAL PRODUCTS

MM5865 Stopwatch Timer	9.00
PC board	7.50
Switches Mom. Pushbutton	.27
3 pos. slide	.25
Encoder HD0165-5	8.95
3 Digit Universal Counter Board Kit Operates 5–16 Volt DC to 5 MHz typ .125" LED display	**10.50**
Voice actuated switch	.50
Paratronics 100A Logic Analyzer Kit	**$224.00**
Model 10 Trigger Expander Kit	$229.00
Model 150 Bus Grabber Kit	$369 00

TRANSFORMERS

8V 300 ma	3.25
12 Volt 300 ma transformer	1.25
12.8V CT 600 ma	3.75
12V 250 ma wall plug	2.95
12V CT 250 ma wall plug	3 50
24V CT 400 ma	3.95
10V 1.2 amp wall plug	4 85
18V 8 amp	12.95

DISPLAY LEDS

MAN1	CA	.270	2.90
MAN3	CC	.125	39
MAN72/74	CA/CA	.300	1.00
DL704	CC	.300	1.25
DL707/DL707R	CA	300	1.00
DL727/728	CA/CC	500	1.90
DL747/750	CA/CC	600	1.95
DL750	CC	.600	1.95
FND359	CC	.357	.70
FND500/507	CC/CA	.500	1.35
FND503/510	CC/CA	500	.90
FND800/807	CC/CA	800	2.20
3 digit Bubble			.60
4 digit bubble			.80
DG8 Fluorescent			1.75
DG10 Fluorescent			1.75
5 digit 14 pin display			1.00
NSN69 9 digit display			.60
7520 Clairex photocells			.39
TIL311 Hex			9.50

COMPUTER GRADE CAPS

1600	mfd	200V	4.50
2000	mfd	45V	2.50
3200		50V	2.50
5500		25V	2.50
5800		40V	3.00
8100		40V	3.00
8100		50V	3.00
7700		50V	2.50
8000		55V	2.50
9400		65V	3.00
10000		20V	2.50
12000		30V	3.50
35000		25V	5.00
55000		25V	5.00
82000		15V	6.00
160000		10V	8.00

QUEST Cosmac Super Elf Computer $106.95

Compare features before you decide to buy any other computer. There is no other computer on the market today that has all the desirable benefits of the **Super Elf** for so little money. The Super Elf is a small single board computer that does many **big** things. It is an excellent computer for training and for learning programming with its machine language and yet it is easily **expanded** with **additional memory, Tiny Basic, Ascii Keyboards, video character generation, etc.**

The **Super Elf** includes a **ROM** monitor for program loading, editing and execution with **SINGLE STEP for program debugging** which is not included in others at the same price. With **SINGLE STEP** you can see the microprocessor chip operating with the **unique Quest** address and data bus displays **before, during** and **after** executing instructions. Also, CPU mode and instruction cycle are shown on several LED indicator lamps.

An RCA 1861 **video graphics** chip allows you to connect to your own TV with an inexpensive video modulator to do graphics and games. There is a **speaker system** included for writing your own music or using many music programs already written. The speaker amplifier may also be used to drive relays for control purposes.

A 24 key **HEX keyboard** includes 16 HEX keys plus **load, reset, run, input, memory protect, memory select, monitor select** and **single step**. Large, on board displays provide output and optional **high and low address**. There is a 44 pin standard connector for PC cards and a 50 pin connector for the **Quest Super Expansion Board**. Power supply and sockets for **all** IC's are included in the price plus a detailed 90 page instruction manual.

Many schools and universities are using the Super Elf as a course of study. OEM's use it for training and research and development.

Remember, other computers only offer Super Elf features at additional cost or not at all. **Compare before you buy. Super Elf Kit $106.95, High address option $8.95, Low address option $9.95. Custom Hardwood Cabinet** with drilled and labelled front panel **$24.95. NiCad Battery Backup Kit $4.95.** All kits and options also come completely assembled and tested.

Questdata, a 12 page monthly software publication for 1802 computer users is available by subscription for $12.00 per year.

Tiny Basic for ANY 1802 System

Cassette $10.00. On ROM Monitor $38.00. Super Elf owners, 30% off. **Object code listing or paper tape with manual $5.50.**
Original ELF Kit Board $14.95.

Super Expansion Board with Cassette Interface $89.95

This is truly an astounding value! This board has been designed to allow **you** to decide how you want it optioned. **The Super Expansion Board comes with 4K of low power RAM** fully addressable anywhere in 64K with built-in memory protect and a **cassette interface**. Provisions have been made for all other options on the same board and it fits neatly into the hardwood cabinet alongside the **Super Elf**. The board includes slots for up to 6K of **EPROM** (2708, 2758, 2716 or TI 2716) and is **fully socketed** ($12.00 value). EPROM can be used for the monitor and Tiny Basic or other purposes.

A 1K Super ROM Monitor $19.95 is available as an on board option in 2708 EPROM which has been preprogrammed with a program loader/editor and error checking multi file cassette read/write software, (relocatible cassette file) another exclusive from Quest. It includes register save and readout, video graphics driver with blinking cursor and block move capability. The Super Monitor is written with subroutines allowing users to take advantage of monitor functions simply by calling them up. Improvements and revisions are easily done with the monitor. If you have the **Super Expansion Board** and **Super Monitor** the monitor is up and running at the push of a button.

Other on board options include **Parallel Input and Output Ports** with **full handshake**. They allow easy connection of an ASCII keyboard to the input port. **RS 232 and 20 ma Current Loop** for teletype or other device are on board and if you need more memory there are two **S-100** slots for static RAM or video boards. A Godbout 8K RAM board is available for $127.95. **Parallel I/O Ports $9.85, RS 232 $4.50, TTY 20 ma I/F $1.95, S-100 $4.50.** A **50 pin connector set** with ribbon cable is available at $12.50 for easy connection between the **Super Elf** and the **Super Expansion Board.**

The **Power Supply** for the Super Expansion Board is a 5 amp supply with + 8v ± 18v + 12v − 5v. Regulated voltages are +5v & +12v $29.95. Deluxe version includes the case at $39.95.

Auto Clock Kit $15.95

DC clock with 4-.50" displays. Uses National MA-1012 module with alarm option. Includes light dimmer, crystal timebase PC boards. Fully regulated, comp. instructs. Add $3.95 for beautiful dark gray case. Best value anywhere.

RCA Cosmac VIP Kit 229.00

Video computer with games and graphics.

Not a Cheap Clock Kit $14.95

Includes everything except case. 2-PC boards. 6-.50" LED Displays. 5314 clock chip, transformer, all components and full instrucs. Green and orange displays also avail. **Same kit w/.80" displays.** Red only. **$21.95**

60 Hz Crystal Time Base Kit $4.40

Converts digital clocks from AC line frequency to crystal time base. Outstanding accuracy. Kit includes: PC board, MM5369, crystal, resistors, capacitors and trimmer.

Digital Temperature Meter Kit

Indoor and outdoor. Switches back and forth. Beautiful. 50" LED readouts. Nothing like it available. Needs no additional parts for complete, full operation. Will measure −100° to +200°F, tenths of a degree, air or liquid. Very accurate. **$39.95**
Beautiful hardwood case w/bezel **$11.75**

NiCad Battery Fixer/Charger Kit

Opens shorted cells that won't hold a charge and then charges them up, all in one kit w/full parts and instructions. **$7.25**

PROM Eraser

Ultraviolet, assembled **$49.95**

Clock Calendar Kit $23.95

CT7015 direct drive chip displays date and time on .6" LEDS with AM-PM indicator. Alarm/doze feature includes buzzer. Complete with all parts, power supply and instructions, less case.

Sinclair 3½ Digit Multimeter

Batt./AC oper. 1mV and .1NA resolution. Resistance to 20 meg. 1% accuracy. Small, portable, completely assem. in case. 1 yr. guarantee. Best value ever! **$59.95**

Video Modulator Kit $8.95

Convert your TV set into a high quality monitor without affecting normal usage. Complete kit with full instructions.

2.5 MHz Frequency Counter Kit

Complete kit less case **$37.50**

30 MHz Frequency Counter Kit

Complete kit less case **$47.75**
Prescaler kit to 350 MHz **$19.95**

1978 IC Update Master Manual

1978 IC Update Master Manual **$24.50.** Complete IC data selector, 2175 pg. Master reference guide. Over 42,000 cross references. Free update service through 1978. Domestic postage $3.50. Foreign $6.00.

Stopwatch Kit $26.95

Full six digit battery operated. 2–5 volts. 3.2768 MHz crystal accuracy. *Times to 59 min., 59 sec., 99 1/100 sec.* Times std., split and Taylor. 7205 chip, all components minus case. Full instructions.

D Connectors RS232

25 Pin Subminiatures

DB25P	2.95	DE9P	1.50
DB25S	3.95	DE9S	1.95
Cover	1.50	DA15P	2.10
RS232 Complete Set	6.50	DA15S	3.10

S-100 Computer Boards

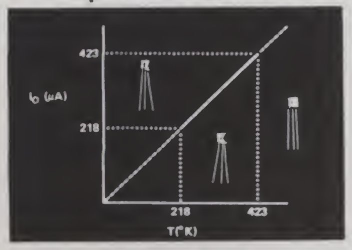
8K Static RAM Kit	$127.00
16K Static RAM Kit	265.00
24K Static RAM Kit	423.00
32K Dynamic RAM Kit	449.00
64K Dynamic RAM Kit	945.00
8K/16K Eprom Kit (less PROMS)	**$89.00**
Video Interface Kit	**$139.00**
Motherboard **$39.** Extender Board	**$8.99**

**TERMS: $5.00 min. order U.S. Funds. Calif residents add 6% tax.
BankAmericard and Master Charge accepted.
Shipping charges will be added on charge cards.**

FREE: Send for your copy of our NEW 1978 QUEST CATALOG. Include 28¢ stamp.

Reader Service—see page 179

ANALOG DEVICES
Two-Terminal IC Temperature Transducer



Two terminal I.C. Temperature Transducer
ANALOG DEVICES AD590J is a two terminal device producing an output current proportional to absolute temperature. Laser triming produces ± 9°C maximum error without external calibration. Calibration can reduce maximum error to only ± 2°C over -55° to +150°C range. Sensitivity is 1uA/°K. Use with +4 to +30V supply as input to digital meter in thermometer applications. Excellent for remote applications due to the very high impedance. Comes in TO-52 metal can.

AD590J....................................$3.49
Specs and Application sheets.................... .80

MC1413P (ULN2003A) Hex Darlington	$ 1.59
THM-6073B TO-220 Heat Sink	5/$ 1.00
MC14411P Baud Rate generator	$11.98
1.843 Crystal for MC14411	$ 4.95
MM57109 Number Cruncher	$18.95
H11F3 Opto-Fet Linear Isolator	$ 1.95
CA3130E Bi MOS OP Amp	$ 1.27
CA3140E Bi MOS OP Amp	$ .57
40673 Popular dual gate FET	$ 1.01
MM5865N Universal Timer I.C.	$ 9.75
CSC 500 MHz prescaler	$59.95
B and K 2800 3½ Digit DVM	$99.95
B and K Dual Tracer Scope Model 1432 (FOB Phoenix)	$750.00

MK5102 (N)-5 Touch Tone Receiver I.C.	$34.95
600ohm to 600ohm C.T. Xfmr	$ 1.95
3.579545 Color Burst Xtal	$ 1.75

Tri-Tek is a Stocking Distributor for factory direct lines of components, tools and equipment. Be sure to get our big catalog!

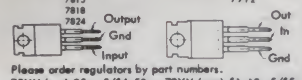
ADEL
ANTRONICS
AP PRODUCTS
AVA
B&K PRECISION
CIRCUIT ASSEMBLY
CONTINENTAL SPECIALTIES CORP.
DATAK
DIXON
EZ HOOK
EDSYN
GRI
HAYDEN
HUNTER
ICO RALLY
IEE
KEYSTONE

3-TERMINAL VOLTAGE REGULATORS
TRI-TEK has become a major source for the 7800 and 7900 series 3-terminal regulators in TO-220 package. These PRIME parts are factory first quality!! At these prices, you can't afford re-tested fall outs. Available in the following voltages. Last 2 digits indicate voltage.

Lowest Prices

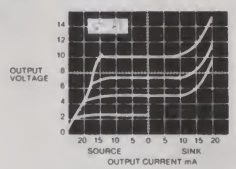
POSITIVE: 7805, 7806, 7808, 7812, 7815, 7818, 7824

NEGATIVE: 7905, 7906, 7908, 7912, 7915, 7918, 7924



Please order regulators by part numbers.
78XX (pos) 95¢, 5/$4.50, 79XX (neg) $1.10, 5/$5

ANALOG DEVICES
Pin Programmable Precision Voltage Reference



AD584 Output Voltage vs. Sink and Source Current

Pin Programmable Precision Voltage Reference
ANALOG DEVICES AD584JH offers pin-programmable selection of four popular output voltages: 10.000V, 7.500V, 5.000V and 2.500V. Laser trimming results in the most flexible monolithic precision reference available. Strobe input allows unit to "turn off" for use in power supply control. Output can sink or source current to greater than 10mA!
Maximum error over full temp range (0-70°c) is ± 30mV. Perfect for use in A-D converters, power supplies, calibrators, etc. 8 pin TO-5 metal can package.

AD584JH..$6.95
Spec and Data Sheets.......................... .60

78P05SC 5V 10A TO-3 Reg	$12.95
21L02-4 (MM2102AN-4L 450nS)	$ 1.44
Z80CPU I.C.	$20.00
Z80ACPU I.C.	$28.00
Z80CTC I.C.	$11.00
Z80ACTC I.C.	$16.00
Z80PIO I.C.	$11.00
Z80APIO I.C.	$16.00
25 Pin RS-232 connector Male	$2.25
25 Pin RS-232 connector Female	$3.35
DB-51226 Hood for RS-232 connector	$ 1.39
D-20418 Screw Lock assembly	$ 1.19
4801 4KX1 Static RAM	$8.95, 8/ $60.00
4804 1KX4 Static RAM	$8.95, 8/ $60.00

MICRO PROCESSOR COMPATIBLE A/D CONVERTERS
National Semi ADC3511 and 3711 are CMOS monolithic A/D converters. Pluse modulation analog-to-digital conversion is used and requires no external precision components except the reference voltage.

Single 5 volt operation. Operation with an isolated supply allows conversion of either polarity. Sign is automatically determined and indicated on the sign pin.

The ADC3511 and 3711 have been designed to provide address BCD data and are intended for use with micro processors and other digital systems. BCD digits are selected on demand via 2 Digit Select inputs. Digit Select inputs are latched. Start conversion input and conversion complete output are included on both types.

FEATURES
- .Single 5V Supply
- .ADC3511 3½ digits (0 to ± 1999 counts)
- .ADC3711 3 3/4 digits (0 to ± 3999 counts)
- .Addressed BCD Outputs
- .Easily interfaced to micro processor
- .TTL compatible

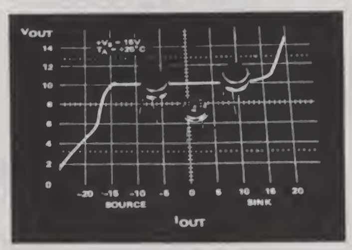
ADC-3511 (3-1/2 digit)	$12.95
ADC-3711 (3-3/4 digit)	$14.95
Spec for both parts	.80

POWER OP AMP
250mA output current capability. Operates on as low as 3V. Input parameters are programmable for system optimizing. Electronic shut down allows output to float. Packaged in 8 pin mini-dip.

LM13080N.......................... $1.94
Specs and applications...... .60

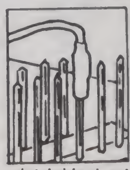
TRI-TEK

ANALOG DEVICES
High Precision 10 Volt IC Reference



High Precision 10Volt IC Reference
ANALOG DEVICES AD581J is monolithic I.C. which produces a precise 10V (± 13.5mV) with 12 to 40V input. Ideal for many A-D converter applications as well as calibrators, power supplies, etc. 3 terminal package is as easy to use as an ordinary regulator. Can be used as precision current source, can be buffered for very large current outputs. Use as 2 terminal device produces a precision zener.

AD581J (3 terminal TO-5).....................$4.98
Specs and Application sheets.................... .60



Jumper Kits for .025 Square Posts. All material for making jumpers for quick circuit changes and prototyping. Use for breadboarding, trouble shooting, field modifications. Fits standard IC socket wire/wrap posts. Excellent wiping action on gold plated box contacts.
Kit contain 10 box contacts, heat shrinkable sleeving, and 5 feet of wire plus instruction sheet.

JCK-5101....(5 double jumpers)$2.75, 4 kits/$10.00

uDP416 16K Dynamic RAM (200nS)	8/ $144.00
uDP416 16K Dynamic RAM (300nS)	8/ $128.00
6502 uProcessor	$10.95

VECTOR
LMB
MOLEX (WALDOM)
OK TOOL
PANAVISE
PCP
POSITRONICS
HOWARD SAMS
H.H. SMITH
SPECO
TAB
TEXAS INSTRUMENTS
THERMALLOY
UNGAR
WALDOM
WAHL
WELCON

Dozens of others-too many to list here and adding more each issue.

Yours for the Asking

TL497C SWITCHING REGULATOR
Switching regulator chip can convert from one voltage to another-either up or down!! Use this one to get positive or negative 12V from your +5V supply for those MOS parts. Starting with anywhere from +4.5V to +12V you can get from +30V down to -25V out. Adjustable output at up to 500mA. We are offering the chip plus a miniature inductor needed for the converter and spec sheet for only $9.50.

TL497C + inductor+spec............$9.50

TRI-TEK
7808 North 27th Avenue
Phoenix, Arizona 85021
(602) 995-9352

T1

Charge card telephone orders ($20 min.) will be accepted 9-5:30 P.M. except weekends.
Telephone (602) 995-9352. No collect calls please.

- For premium shipping (first class, special handling, etc.) add extra. Excess will be refunded
- Please give street address for UPS shipping when possible
- C O D NO parcel post C.O.D. UPS C.O.D. Add 85¢ to order
- Any correspondence not connected with your order, please use separate sheet and include SASE for reply
- Claims for shortages or damaged material must be made within 10 days of receipt of package
- Prices are subject to change without notice

(U. S. DOLLARS ONLY

- Any refunds will be by check, not credit vouchers
- If we should be temporarily out of stock on any item, it will be placed on back order. If we cannot ship in 30 days, you will be notified of the expected shipping date and furnished with a postage paid card with which to cancel your order if desired
- Terms, Check, money order, credit card. Net 30 days to rated firms, schools and government agencies.

Claims for defective material must be made within 60 days of receipt of parcel. Claim must include invoice number, date & defective parts.

ARIZONAN S ADD 5% TAX

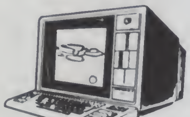
Shipping and Handling
U.S. and CANADA ADD 5%
Other countries add 15%

(MINIMUM ORDER U.S. $10.00)
(FOREIGN $15.00)

ADVANCED COMPUTER PRODUCTS
A38 –NOW IN STOCK!

COMPUTER SPECIALS

. . . and it's COLOR
"The Compucolor II"
. . . a personal colorgraphics system for the modern computer man . . .



- ★ Color Graphics 13" Color CRT
- ★ Proven 8080A CPU System
- ★ 16K Extended Disk Basic
- ★ Up to 117* Key Keyboard
- ★ Up to 32K* RAM
- ★ Minidisk Drive 51.2K Bytes/Side

*option

Model 3 w/8K, 72 Key Keyboard, RS232 . . . . $1495.00
Model 4 w/16K, 72 Key Keyboard, RS232 . . . . $1795.00
Model 5 w/32K, 72 Key Keyboard, RS232 . . . . $2395.00
Options: 101 Key Keyboard . . . . Add $150.00
117 Key Keyboard . . . . Add $225.00
Formatted Diskettes. . . . . 2/$19.95
Programmed Diskettes $19.95
Diskette Library Inc. Hangman, Othello, Math, Chess, Startrek, Blackjack, Cubic Tic Tac Toe, Finance Vol. I, Finance Vol. II, Bonds and Securities, Assembler, Text Editor, Personal Data Base.

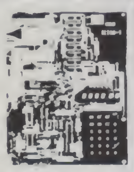
EXIDY SORCERER™

$895 w/8K
$1150 w/16K (Stock)
$1395 w/32K (Stock)

User programmable or use cartridges. Combines the desirable features of the PET, APPLE and TRS-80 into a complete expandable computer system. If you are a comparison shopper send for comparison chart.

★ INCLUDES:
Keyboard & enclosure totally assembled
90 day Warranty
MICROSOFT BASIC
Video & Cassette Cable
Complete Documentation
★★ S100 Expansion Module . . . . . Add $299.00
★★ Cassette recorder . . . . Add $44.95
★★ Sanyo 9" Monitor . . . . Add $169.95

KIM-1



Now only $179.00

The KIM-1 6502 Standard Computer System now available off the shelf. Write for complete details and software package.
★★ Power Supply . . . . Add $59.95
★★ Cassette Recorder . . . . Add $44.95
★★ Sanyo 9" Monitor . . . . Add $169.95

SYM-1

"The Ultimate in Single Board Low-Cost Computers" In Stock
Reg. $269.00
Now $249.00 IN STOCK

- ★ KIM-1 Compatible
- ★ 4K ROM Monitor
- ★ 1K Bytes 2114 RAM
- ★ 65K Memory Expansion
- ★ User EPROM 2716

★★ Power Supply . . . . Add $59.95
★★ Cassette Recorder . . . . Add $44.95
★★ Sanyo 9" Monitor . . . . Add $169.95

School & group discounts available.

AIM

NEW FROM ROCKWELL INTERNATIONAL

Singleboard Computer

- ★ On Board 20 column alphanumeric printer
- ★ Alphanumeric 20 character display
- ★ Terminal style Keyboard 54 Keys
- ★ 6502 based CPU

w/1K RAM $375.00
w/4K RAM $450.00
Assembler ROM . . . Add $85.00
BASIC Interpreter in ROM . . . . Add $100.00

RCA COSMAC VIP

NEW LOW PRICE **$249.00**
Assembled
Regular price $299.95
w/Sanyo 9" Monitor . . . . Add $169.95

"Now you can afford to experiment using RCA's fine 1802 CMOS CPU."

BALLY ARCADE

Z-80 based expandable to 48K
Color Display BASIC available in ROM Cartridge
8K ROM / 4K RAM **$299.95**
Order Bally Cassettes @$19.95 ea. except 3001 @$24.95, 2001 Zzap & Dodgem, 2002 Sea Wolf & Missle, 2003 Attack & Red Baron, 3001 Baseball, 4001 Bingo Math & Speed Math, 4002 Letters & Crosswords, 5002 Blackjack & Poker

NORTH STAR HORIZON

Now in stock North Star Z-80 based high-performance computer.

- ★ Z-80 Processor
- ★ Motherboard
- ★ 2 Serial +1 Parallel Port
- ★ 16K RAM

Horizon I . . . . $1439.00 Kit
Horizon II. . . . $1799.00 Kit

PRINTERS

OKIDATA Model 110 w/tractor w/RS232 . . . . $1675.00
OKIDATA Model 22 w/tractor w/RS232 . . . . $2705.00
DECWRITER II w/RS232 10-30 cps $1675.00
DIABLO 1620-3 w/tractor feed, w/Keyboard. $3255.00
DIABLO 1610-3 w/tractor feed $2995.00
IPSI 1622-3 w/tractor feed (diablo compatible). $2995.00
IPSI 1612-3 w/tractor feed (diablo compatible). $2820.00
Centronics 761 (KSR) $1595.00
Centronics Micro S-1 . . . . $525.00
Centronics Micro P-1 . . . . $475.00
Centronics 761RO $1495.00
Centronics 779 w/tractor feed. . . $1195.00
Teletype Model 43 w/RS232 . . . $1199.00
TI Model 745 w/upper & lower cs $1975.00
TI Model 810. . . . . $1895.00

FLOPPY DISK'S

Shugart SA400 Minifloppy Disk Drive $295.00
Shugart SA450 Dual Density Minifloppy $375.00
Shugart SA800/801 Diskette Storage Drive $495.00
Shugart SA850/851 Double-Sided Drive $625.00
MFE Mayflower Model 751 Double Density. $730.00
Persci Model 277 Dual Diskette Storage Drive $1230.00
Pertec FD200 Miniflop Disk Drive $299.00
Pertec Model 88 8" Drive $495.00
Calcomp Model 143M Dual Density Drive $625.00
North Star MDS Minifloppy w/S-100 . . . . $599.00

TERMINALS

SOROC IQ 120. . . . . $895.00
LEAR ADM3 Assembled $829.95
LEAR ADM3 Kit $739.95
SOROC IQ 140 . . . . $1495.00
Sanyo 9" Monitor $159.95
Sanyo 15" Monitor. $209.95
Motorola 12" Monitor w/o chassis . $219.95
Hazeltine 1400 . . . . $750.00
Hazeltine 1500 . . . . $1095.00
Hazeltine 1510 . . . . $1295.00

KEYBOARDS

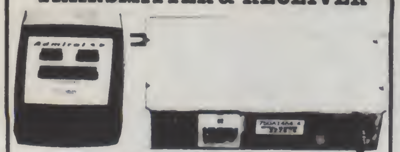
GEO Risk Model 756 ASCII 56 Key Assembled $67.95
Metal case for Model 756. . . . . $27.00
Clare Pender 62 Key ASCII w/26 Pin and 34 Pin Connector (new surplus supply limited). $54.95
63 Key Unencoded Keyboard $32.95
10 Key Hexpad Unencoded. $11.95

ADVANCED COMPUTER PRODUCTS DEPT K
P. O. Box 17329 • Irvine, California 92713
1310 B E. Edinger • Santa Ana, California 92705
TWX/TELEX 910 595 1565 • TEL (714) 558-8813

All prices FOB Santa Ana. Please add shipping/handling. For credit card purchase contact for quotation. Complete FREE 1979 Catalogue – send 25¢ for postage. Pricing subject to change without notice.

Reader Service—see page 179

REMOTE CONTROL TRANSMITTER & RECEIVER

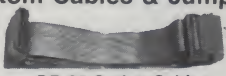


- CAN BE USED AS REMOTE CONTROL FOR TV
- USE IT FOR YOUR OWN GARAGE DOOR OPENER
- TRANSMITS BETWEEN 38KHz AND 41.5KHz
- THOUSANDS OF USES FOR REMOTE CONTROL APPLICATIONS
- TRANSMITTER USES A 1.5V BATTERY
- SCHEMATIC INCLUDED

$19.95

Custom Cables & Jumpers

NEW NEW



DB 25 Series Cables

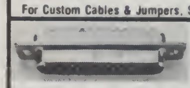
Part No.	Cable Length	Connectors	Price
DB25P-4-P	4 Ft.	2-DP25P	$15.95 ea.
DB25P-4-S	4 Ft.	1-DP25P/1-25S	$16.95 ea.
DB25S-4-S	4 ft.	2-DP25S	$17.95 ea.

Dip Jumpers

DJ14-1	1 ft.	1-14 Pin	$1.59 ea.
DJ16-1	1 ft.	1-16 Pin	1.79 ea.
DJ24-1	1 ft.	1-24 Pin	2.79 ea.
DJ14-1-14	1 ft.	2-14 Pin	2.79 ea.
DJ16-1-16	1 ft.	2-16 Pin	3.19 ea.
DJ24-1-24	1 ft.	2-24 Pin	4.95 ea.

For Custom Cables & Jumpers, See JAMECO 1979 Catalog for Pricing

CONNECTORS



25 Pin-D Subminiature

DB25P (as pictured)	PLUG (Meets RS232)		$2.95
DB25S	SOCKET (Meets RS232)		$3.50
DB51226-1	Cable Cover for DB25P or DB25S		$1.75

PRINTED CIRCUIT EDGE-CARD

156 Spacing-Tin-Double Read-Out — Bifurcated Contacts — Fits .054 to .070 P.C. Cards

15/30	PINS (Solder Eyelet)		$1.95
18/36	PINS (Solder Eyelet)		$2.49
22/44	PINS (Solder Eyelet)		$2.95
50/100 (.100 Spacing)	PINS (Wire Wrap)		$6.95
50/100 (.125 Spacing)	PINS (Wire Wrap)	R681-1	$6.95

MINI-BUZZER

- Thousands of applications
- Operates on +6 to +9 VDC
- Output Frequency 800 Hz
- Draws only 15 MA @ 600 ohms
- Size: 1¼" x ⅝"

PART NO. MB-1

1-24 $1.95 ea
25-49 $1.75 ea
50-99 $1.49 ea

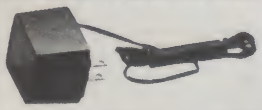
SONALERT® AUDIBLE SIGNAL DEVICE

"Use as a warning Device or Audible Reminder"

- Turns on and off with low power transistor, SCR or IC
- Can be battery operated 6-28 volts
- Solid State - No moving parts.
- Panel mounts in 1.25" round hole
- Black Plastic case-includes MTG nut.
- **Operating Volts (DC):** 6 (Min) – 28 (Max)
- **Current MA:** 3 (Min) – 14 (Max)
- **dBA:** 68 (Min) – 80 (Max)
- **Frequency:** 2900Hz ± 500

#SC628
$7.50 each

AC Wall Transformer



Ideal for use with clocks, power supplies or any other type of AC application.

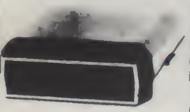
Part No.	Input	Output	Price
AC 250	117V/60Hz	12 VAC 250mA	$3.95
AC 500	117V/60Hz	12 VAC 500mA	$4.95

Regulated Power Supply

- Uses LM 309K
- Heat sink provided
- P.C. board construction
- Provides a solid 1 amp @ 5V
- Includes components, hardware and instructions
- Sizes: 3-1/2" x 5" x 2" high

JE200 $14.95

INSTRUMENT/CLOCK CASE



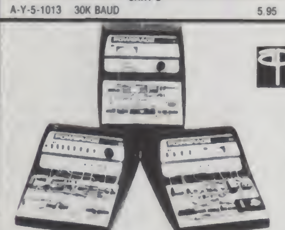
This case is an injection molded unit that is ideal for uses such as DVM, COUNTER, or CLOCK cases. It has dimensions of 4½" in length by 4" in width by 1-9/16" in height. It comes complete with a red bezel.

PART NO: IN-CC $3.49 each

MICROPROCESSOR COMPONENTS

8080A/8080A SUPPORT DEVICES		
8080A	CPU	$9.95
8212	8-Bit Input/Output	3.25
8214	Priority Interrupt Control	5.95
8216	Bi-Directional Bus Driver	3.49
8224	Clock Generator/Driver	3.95
8226	Bus Driver	3.49
8228	System Controller/Bus Driver	5.95
8238	System Controller	5.95
8251	Prog. Comm. I/O (USART)	7.95
8253	Prog. Interval Timer	14.95
8255	Prog. Periph. I/O (PPI)	9.95
8257	Prog. DMA Control	19.95
8259	Prog. Interrupt Control	19.95
6800/6800 SUPPORT DEVICES		
MC6800	MPU	$14.95
MC6802CP	MPU with Clock and Ram	24.95
MC6810API	128X8 Static Ram	5.95
MC6821	Periph. Inter. Adapt (MC6820)	7.49
MC6828	Priority Interrupt Controller	12.95
MC6830L8	1024X8 Bit RDM (MC68A30-8)	14.95
MC6850	Asynchronous Comm. Adapter	7.95
MC6852	Synchronous Serial Data Adapt.	9.95
MC6860	0-600 bps Digital MODEM	12.95
MC6862	2400 bps Modulator	14.95
MC6880A	Quad 3-State Bus. Trans. (MC8T26)	2.25
MICROPROCESSOR CHIPS—MISCELLANEOUS		
Z80(780C)	CPU	$19.95
Z80A(780-1)	CPU	24.95
CDP1802	CPU	19.95
2650	MPU	19.95
8035	8-Bit MPU w/clock, RAM, I/O lines	19.95
P8085	CPU	19.95
TMS9900JL	16-Bit MPU w/hardware, multiply & divide	49.95
SHIFT REGISTERS		
MM500H	Dual 25 Bit Dynamic	$.50
MM503H	Dual 50 Bit Dynamic	.50
MM504H	Dual 16 Bit Static	.50
MM506H	Dual 100 Bit Static	.50
MM510H	Dual 64 Bit Accumulator	.50
MM5016H	500/512 Bit Dynamic	.89
2504T	1024 Dynamic	3.95
2518	Hex 32 Bit Static	4.95
2522	Dual 132 Bit Static	2.95
2524	512 Static	.99
2525	1024 Dynamic	2.95
2527	Dual 256 Bit Static	2.95
2528	Dual 250 Static	4.00
2529	Dual 240 Bit Static	4.00
2532	Quad 80 Bit Static	2.95
2533	1024 Static	2.95
3341	Fifo	6.95
74LS670	4X4 Register File (TriState)	1.95
UART'S		
A-Y-5-1013	30K BAUD	5.95

MICROPROCESSOR MANUALS			
M-Z80	User Manual		$7.50
M-CDP1802	User Manual		7.50
M-2650	User Manual		5.00
ROM'S			
2513(2140)	Character Generator(upper case)		$9.95
2513(3021)	Character Generator(lower case)		9.95
2516	Character Generator		10.95
MM5230N	2048-Bit Read Only Memory		1.95
RAM'S			
1101	256X1	Static	$1.49
1103	1024X1	Dynamic	.99
2101(8101)	256X4	Static	3.95
2102	1024X1	Static	1.75
21L02	1024X1	Static	1.95
2111(8111)	256X4	Static	3.95
2112	256X4	Static MOS	4.95
2114	1024X4	Static 450ns	9.95
2114L	1024X4	Static 450ns low power	10.95
2114-3	1024X4	Static 300ns	10.95
2114L-3	1024X4	Static 300ns low power	11.95
5101	256X4	Static	7.95
5280/2107	4096X1	Dynamic	4.95
7489	16X4	Static	1.75
74S200	256X1	Static Tristate	4.95
93421	256X1	Static	2.95
UPD414 (MK4027)	4K	Dynamic 16 pin	4.95
UPD416 (MK4116)	16K	Dynamic 16 pin	14.95
TMS4044-45NL	4K	Static	14.95
TMS4045	1024X4	Static	14.95
2117	16,384X1	Dynamic 350ns (house marked)	9.95
MM5262	2KX1	Dynamic	4/1.00
PROM'S			
1702A	2048	FAMOS	$5.95
TMS2516 (2716)	16K*	EPROM(Intel 2716)	49.95
	*Requires single +5V power supply		
TMS2532	4KX8	EPROM	89.95
2708	8K	EPROM	10.95
2716 T.I.	16K**	EPROM	29.95
	**Requires 3 voltages, —5V, +5V, +12V		
5203	2048	FAMOS	14.95
6301-1(7611)	1024	Tristate Bipolar	3.49
6330-1(7602)	256	Open C Bipolar	2.95
82S23	32X8	Open Collector	3.95
82S115	4096	Bipolar	19.95
82S123	32X8	Tristate	3.95
74186	512	TTL Open Collector	9.95
74188	256	TTL Open Collector	3.95
74S287	1024	Static	2.95



NEW!! IN STOCK...

POWERACE

ALL-CIRCUIT EVALUATORS WITH POWER

- 1680 solderless, plug-in tie points, will hold up to 18 14-pin DIP's.
- Breadboard elements accept all DIP sizes, including RTL, DTL, TTL and CMOS devices TO-5's and discretes with leads up to .032" dia.
- All connections to/from switches, indicators, power supplies and meters are made via solderless, plug-in, tie-point blocks on control panels.
- Interconnect with any solid 20 to 30 AWG wire.
- Breadboard elements are mounted on ground planes... ideal for high-frequency and high-speed/low-noise circuits
- Short-circuit-proof fused power supplies.
- Operate on 110 to 130 VAC at 60 Hz
- Space-age compact styling and high-grade components permit convenient, organized and quick prototyping.
- All models are 7.5" wide 11.5" deep and 4.0" high (rear) 0.75" high (front) and weigh approx. 2.5 lbs.

POWERACE 101 -General purpose model for prototyping all types of circuit.
#923101 $ 84.95

POWERACE 102 -Complete digit prototyping lab with built-in logic probe
#923102 $114.95

POWERACE 103 -Triple-output power supply for prototyping both linear and digit circuits.
#923103 $124.95

B+K PRECISION

3½-Digit Portable DMM

- Overload Protected
- 3 high LED Display
- Battery or AC operation
- Auto Zeroing
- 1mv 1Va 0.1 ohm resolution
- Overange reading
- 10 meg input impendence
- DC Accuracy 1% typical

Ranges: DC Voltage - 0-1000V
AC Voltage 0-1000V
Freq. Response 50-400 HZ
DC/AC Current 0-100mA
Resistance 0-10 meg ohm
Size 6.4" x 4.4" x 2"

Model 2800
$99.95

Comes with test leads, operating manual and spare fuse

Accessories:

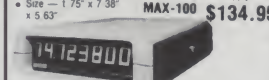
AC Adapter BC-28	$9.00
Rechargeable Batteries BP-26	20.00
Carrying Case LC-28	7.50

CSC CONTINENTAL SPECIALTIES CORPORATION

100 MHz 8-Digit Counter

- 20 Hz-100 MHz Range
- 6" LED Display
- Crystal-controlled timebase
- Fully Automatic
- Portable — completely self-contained
- Size — 1.75" x 7.38" x 5.63"
- Four power souces, i.e. batteries, 110 or 220V with charger 12V with auto lighter adapter and external 7.2-10V power supply

MAX-100 $134.95



ACCESSORIES FOR MAX 100:

Mobile Charger Eliminator
use power from car battery — **Model 100 — CLA $3.95**

Charger/Eliminator
use 110 V AC — **Model 100 — CAI $9.95**

New CSC Mini-Max 6 Digit 50MHz Frequency Counter

- Guaranteed frequency range of 100 Hz to 50 MHz
- Full 6 digit display with antiglare window
- Fully automatic-range, polarity, slope, trigger, input level switching not required.
- Lead-zero blanking—All zeros to the left of the first non-zero digit are blanked. Kilo Hertz and Mega Hertz decimal points automatically light up when the unit is turned on.
- Built in input overvoltage protection
- Use 9V Battery or 110/220V power.
- Complete with mini antenna.
- Lightweight — Only 8oz.

MINI-MAX $89.95

Accessories For Mini-Max

Part No.	Description	Price
MM-A4	Antenna	$ 3.95
MM-C5	Carrying case	5.95
MM-IPC	Input cable with clip leads	3.95
MM-AC2	110V adapter	9.95
MM-AC3	220V adapter	9.95

$5.00 Minimum Order — U.S. Funds Only
California Residents — Add 6% Sales Tax
Spec Sheets — 25¢
1979 Catalog Available—Send 41¢ stamp

NEW 1979 Catalog

Jameco ELECTRONICS

a division of JAMES ELECTRONICS of California

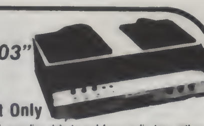
✓ J1

PHONE ORDERS WELCOME (415) 592-8097

MAIL ORDER ELECTRONICS — WORLDWIDE
1021 HOWARD AVENUE, SAN CARLOS, CA 94070

ADVERTISED PRICES GOOD THRU FEBRUARY

The Incredible *"Pennywhistle 103"*



$139.95 Kit Only

The Pennywhistle 103 is capable of recording data to and from audio tape without critical speed requirements for the recorder and it is able to communicate directly with another modem and terminal for telephone "hamming" and communications. In addition, it is free of critical adjustments and is built with non-precision, readily available parts.

Data Transmission Method	Frequency-Shift Keying, full-duplex (half-duplex selectable)
Masimum Data Rate	300 Baud
Data Format	Asynchronous Serial (return to mark level required between each character).
Receive Channel Frequencies	2025 Hz for space, 2225 Hz for mark
Transmit Channel Frequencies	Switch selectable. Low (normal) = 1070 space, 1270 mark, High = 025 space, 2225 mark
Receive Sensitivity	46 dbm accoustically coupled
Transmit Level	15 dbm nominal. Adjustable from −6 dbm to −20 dbm
Receive Frequency Tolerance	Frequency reference automatically adjusts to allow for operation between 1800 Hz and 2400 Hz.
Digital Data Interface	EIA RS-232C or 20 mA current loop (receiver is optoisolated and non-polar).
Power Requirements	120 VAC, single phase, 10 Watts
Physical	All components mount on a single 5" by 9" printed circuit board. All components included.

Requires a VOM, Audio Oscillator, Frequency Counter and/or Oscilloscope to align

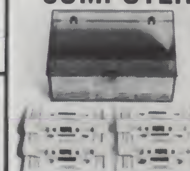
TRS-80 16K Conversion Kit

Expand your 4K TRS-80 System to 16K. Kit comes complete with:
* 8 each UPD416 (16K Dynamic Rams)
* Documentation for conversion

TRS-16K $115.00

Special Offer - Order both your TRS-16K and the Sup'R' MOD II Interface kit together (retail value $144.95) **for only $139.95**

COMPUTER CASSETTES



- 6 EACH 15 MINUTE HIGH QUALITY C-15 CASSETTES
- PLASTIC CASE INCLUDED 12 CASSETTE CAPACITY
- ADDITIONAL CASSETTES AVAILABLE #C-15-$2.50 ea

CAS-6
$14.95
(Case and 6 Cassettes)

SUP 'R' MOD II

UHF Channel 33 TV Interface Unit Kit

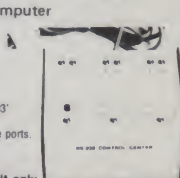
Wide Band B/W or Color System

★ Converts TV to Video Display for home computers, CCTV camera, Apple II, works with Cromeco Dazzler, SOL-20, TRS-80, Challenger, etc.
★ MOD II is pretuned to Channel 33 (UHF).
★ Includes coaxial cable and antenna transformer.

MOD II $29.95 Kit

RS-232 CONTROL CENTER

Plug in your modem, computer prom programmer, terminal, printer, etc. and selectively control data flow.



- Same Contour as "Pennywhistle 103"
- Totally self-contained
- Includes 2 master ports and 3 slave ports.

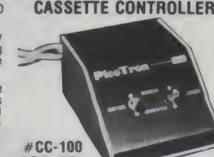
PART NO.
RS-232CC $89.95 kit only

CASSETTE CONTROLLER

Ideal for use with the TRS 80 and others.

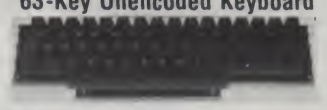
"Plug/Jack interface to any computer system requiring remote control of cassette functions"

The CC100 controls cassette motor functions, monitors tape location with its internal speaker and requires no power. Eliminates the plugging and unplugging of cables during computer loading operation from cassette.



#CC-100
$29.50

63-Key Unencoded Keyboard



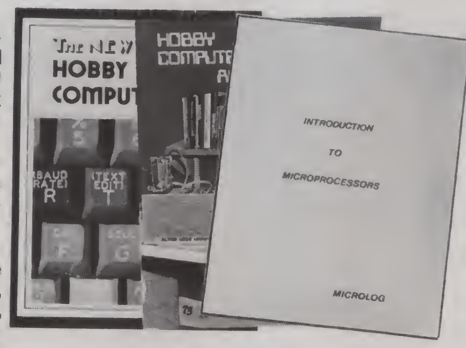
This is a 63-key, terminal keyboard newly manufactured by a large computer manufacturer. It is unencoded with SPST keys, unattached to any kind of PC board. A very solid molded plastic 13 x 4" base suits most application. IN STOCK **$29.95/each**

Hexadecimal Unencoded Keypad

19-key pad includes 1-10 keys, ABCDEF and 2 optional keys and a shift key. **$10.95/each**

DEALER DIRECTORY

KB BOOK NOOK



● **HOBBY COMPUTERS ARE HERE!**—BK7322—If you want to come up to speed on how computers work . . . hardware and software . . . this is an excellent book. It starts with the fundamentals and explains the circuits, the basics of programming, along with a couple of TVT construction projects, ASCII-Baudot, etc. This book has the highest recommendations as a teaching aid for newcomers. $4.95.*

● **THE NEW HOBBY COMPUTERS**—BK7340—This book takes it from where "**Hobby Computers Are Here!**" leaves off, with chapters on Large Scale Integration, how to choose a microprocessor chip, an introduction to programming, low cost I/O for a computer, computer arithmetic, checking memory boards, a Baudot monitor/editor system, an audible logic probe for finding those tough problems, a ham's computer, a computer QSO machine . . . and much, much more! $4.95*

● **INTRODUCTION TO MICROPROCESSORS**—BK1032—by Charles Rockwell of MICROLOG is an ideal reference for the individual desiring to understand the hardware aspects of microprocessor systems. Describes the hardware details of computer devices in terms the beginner can understand, instead of treating the micro chip as a "black box." Specific systems are not described and programming is only briefly discussed. $17.50 US and Canada, $20 elsewhere.*

● **BASIC NEW 2ND EDITION**—BK1081—by Bob Albrecht. Self-teaching guide to the computer language you will need to know for use with your microcomputer. This is one of the easiest ways to learn computer programming. $4.95.*

● **SCIENTIFIC RESEARCH INSTRUMENTS' BASIC SOFTWARE LIBRARY** is a complete do-it-yourself kit. Written in everybody's BASIC immediately executable in ANY computer with 4K, no other peripherals needed. **Volume I** contains business and recreational programs and is 300 pages. **Volume II** is 260 pages and contains math, engineering, statistics and plotting programs. **Volume III** contains money managing, advanced business programs such as billing, A/R, inventory, payroll, etc. **Volume IV** contains general purpose programs like loans, rates, retirement, plus games: Poker, Enterprise (take charge while Capt. Kirk is away), Football and more! **Volume V** is filled with experimenter's programs including games, pictures and misc. problems like "logic." Volume I—LB1002—& Volume II—LB1003—$24.95* each, Volume III—LB1004—$39.95*, Volume IV—LB1005—& Volume V—LB1006—$9.95* each, Volume VI—LB1007—$49.95*, Volume VII—LB1008—$39.95.*

● **MICROPROCESSOR INTERFACING TECHNIQUES**—BK1037—by Austin Lesea & Rodnay Zaks will teach you how to interconnect a complete system and interface it to all the usual peripherals. It covers hardware and software skills and techniques, including the use and design of model buses such as the IEEE 488 or S100. $9.95*

● **TTL COOKBOOK**—BK1063—by Donald Lancaster. Explains what TTL is, how it works, and how to use it. Discusses practical applications, such as a digital counter and display system, events counter, electronic stopwatch, digital voltmeter, and a digital tachometer. $8.95.

● **CMOS COOKBOOK**—BK1011—by Don Lancaster. Details the application of CMOS, the low power logic family suitable for most applications presently dominated by TTL. Required reading for every serious digital experimenter! $9.95.*

● **TVT COOKBOOK**—BK1064—by Don Lancaster, describes the use of a standard television receiver as a microprocessor CRT terminal. Explains and describes character generation, cursor control and interface information in typical, easy-to-understand Lancaster style. $9.95.

● **THE "COMPULATOR" BOOK—BUILDING SUPER CALCULATORS & MINICOMPUTER HARDWARE WITH CALCULATOR CHIPS**—BK1012—by R. P. Haviland, provides ideas, design info and printed circuit boards for calculator chip projects, measure time, tie in with a Teletype to create a virtually infinite memory system, and countless other functions. $7.95.*

● **BASIC COMPUTER GAMES**—BK1074—Okay, so once you get your computer and are running in BASIC, then what? Then you need some programs in BASIC, that's what. This book has 101 games for you, from very simple to real buggers. You get the games, a description of the games, the listing to put in your computer and a sample run to show you how they work. Fun. Any one game will be worth more than the price of the book for the fun you and your family will have with it. $7.50.*

● **WHAT TO DO AFTER YOU HIT RETURN**—BK1071—PCC's first book of computer games . . . 48 different computer games you can play in BASIC . . . programs, descriptions, many illustrations. Lunar Landing, Hammurabi, King, Civel 2, Qubic 5, Taxman, Star Trek, Crash, Market, etc. $8.00.*

● **KILOBAUD CODING SHEETS** Kilobaud's dozen or so programmers wouldn't try to work without these handy scratch pads which help prevent the little errors that can cost hours and hours of programming time. Available for programming is **Assembly/Machine Language** (PD1001) which has columns for address, instruction (3 bytes), source code (label, op code, operand), and comments; and for **BASIC** (PD1002) which is 72 columns wide. 50 sheets to a pad. $2.39.*

* Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to: Kilobaud Book Department ● Peterborough NH 03458. Be sure to include check or detailed credit card information. For orders under $10.00, add $1.00 shipping and handling.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

KB BOOK NOOK

● **HOW TO BUILD A MICROCOMPUTER—AND REALLY UNDERSTAND IT**—BK7325—by Sam Creason. The electronics hobbyist who wants to build his own microcomputer system now has a practical "How-To" guidebook. Sam Creason's book is a combination technical manual and programming guide that takes the hobbyist step-by-step through the design, construction, testing and debugging of a complete microcomputer system. This book is must reading for anyone desiring a true understanding of computer systems. $9.95.*

● **PROGRAMMING IN PASCAL**—BK1140—by Peter Grogono. The computer programming language PASCAL was the first language to embody in a coherent way the concepts of structured programming which had been defined by Edsger Dijkstra and C.A.R. Hoare. As such it is a landmark in the development of programming languages. PASCAL was developed by Niklaus Wirth in Zurich; it is derived from the language ALGOL 60 but is more powerful and easier to use. PASCAL is now widely accepted as a useful language that can be efficiently implemented, and as an excellent teaching tool. $9.95.*

● **MICROPROCESSOR LEXICON—ACRONYMS AND DEFINITIONS**—BK1137—Compiled by the staff of SYBEX is a convenient reference in pocket size format. Sections include acronyms and definitions, part numbers and their definitions, S-100 signals, RS232 signals, IEEE 499 signals, microcomputer and microprocessors, JETDS summary (military) and a code conversion table. $2.95.*

SPECIAL PRE-PUBLICATION OFFER

(orders will be filled on a first-in, first-out basis)

● **INSTANT BASIC**—BK1131—by Jerald R. Brown. For the personal computer enthusiast or the user of DEC's BASIC PLUS language, here, finally, is a new book to teach you BASIC. It teaches BASIC to beginners using interesting programming ideas and applications that will be easily understood by the home computer programmer. BASIC PLUS users know that the two languages are very similar, so this book can be used by them as well. This is an "active participation" workbook, designed to be used with your home computer so you can learn by doing! $6.00.*

● **YOUR HOME COMPUTER**—BK1172—by James White is an introduction to the world of personal microcomputing. Written in clear and understandable language, this book tells you everything you want to know about home computing and gives the computer novice a painless introduction to microcomputer technology and terminology, beginning with what computers are and how they work.

It provides information about home computer kits, guidelines for selecting and building your own microcomputer, how to use your home computer and what you can do with it, lists of computer stores, clubs, periodicals, and answers to many more of your questions about microcomputers and the jargon surrounding the personal computing scene today. $6.00.*

KB BOOK NOOK ORDER FORM

Qty.	Cat.#	Description	Unit Price	Total

Add $1 shipping & handling for orders under $10: ______

Merchandise total: ______

SUBSCRIBE!

☐ Kilobaud ☐ One year—KB: $18* ☐ 3 years—Kilobaud: $45*

☐ 73 ☐ One year—73: $18* ☐ 3 years—73: $45*

☐ Life Sub (73/KB) $195** ☐ New Sub ☐ Renewal

*U.S. & Canada only **U.S. only

Subscription total: ______

Enclosed______ ☐ Check ☐ Money order

Total subscription &/or merchandise: ______

Bill my: ☐ Visa ☐ Master Charge ☐ American Express

Credit card # ____________ Expiration date______

Signature ____________

Name ____________

Address ____________

City ____________ State______ Zip______

kilobaud
peterborough nh 03458

For Toll Free Ordering : (800) 258-5473

(coupon expires in 60 days)

K 2A

FOR CUSTOMER SERVICE CALL (603) 924-6132

kilobaud MICROCOMPUTING

INDEX TO ADVERTISERS

*Reader Service inquiries not honored. Please contact advertiser directly.

READER SERVICE

receive more informa-
n from any of the adver-
ers in this issue of
obaud, postage-paid
urn cards are provided
e for your convenience.
you wish to hear from
e or several advertisers,
er to the ad itself. You
l find numbers near the
o of each advertiser.
ch represents the adver-
er's individual **Reader**
rvice number. Circle the
responding numbers on
e of the cards on this
ge, include your name,
dress & zip and drop in
mailbox. In four to six
eks you'll hear from the
vertiser directly.

kilobaud MICROCOMPUTING

Kilobaud INSTANT INFORMATION CARD

ou're in a real hurry to
ar from one of our
vertisers (some of us
t can't wait), we have
luded an **Instant Infor-**
tion Card for your use.
nply include your name
d full address on the re-
est card. Fill out the
nt of the card with the
ecific advertiser's name
d address, include a
mp, and drop it in the
il. You should receive
ormation within two to
ee weeks.

READER SERVICE REPLY CARD

Return this card to receive full information on the many products advertised in this issue. Limit: 25 requests. ☐ Subscriber ☐ Newsstand. This card is valid until March 31, 1979.

A22	C36	C98	C112	D42	F8	I24	J13	M44	M77	O1	P38	R24	S61	T36	W13
A38	C64	C100	C113	D43	F10	I28	K14	M49	M78	O2	P39	R26	S73	T41	W16
A66	C73	C102	C114	D44	F11	I30	K15	M54	N7	O6	P43	R28	S74	T46	W20
A71	C80	C103	D20	E18	G4	I32	L19	M64	N9	O9	P46	R29	S75	T47	W21
A75	C81	C104	D28	E29	G24	I33	L20	M65	N12	O10	Q3	S6	S76	T49	X2
A92	C86	C107	D32	E34	H25	J1	L21	M67	N13	O11	Q8	S42	T1	T51	73
B33	C87	C108	D37	E36	H29	J4	M17	M70	N15	P7	R7	S51	T11	V17	kilobaud
B35	C94	C109	D40	E37	I9	J6	M30	M73	N16	P9	R8	S52	T26	V19	
B43	C97	C111	D41	E38	I22	J12	M32	M74	N17	P21	R12	S55	T28	V20	

This month's ARTICLE WINNER (title page number): ____________

Name ______________________________

Address ______________________________

City ____________________ State ______ Zip ____________

KB 2/79

kilobaud MICROCOMPUTING
peterborough nh 03458

Place Stamp Here

INSTANT INFORMATION

____
Advertiser

____
Street or box number

____
City State Zip

READER SERVICE REPLY CARD

Return this card to receive full information on the many products advertised in this issue. Limit: 25 requests. ☐ Subscriber ☐ Newsstand. This card is valid until March 31, 1979.

A22	C36	C98	C112	D42	F8	I24	J13	M44	M77	O1	P38	R24	S61	T36	W13
A38	C64	C100	C113	D43	F10	I28	K14	M49	M78	O2	P39	R26	S73	T41	W16
A66	C73	C102	C114	D44	F11	I30	K15	M54	N7	O6	P43	R28	S74	T46	W20
A71	C80	C103	D20	E18	G4	I32	L19	M64	N9	O9	P46	R29	S75	T47	W21
A75	C81	C104	D28	E29	G24	I33	L20	M65	N12	O10	Q3	S6	S76	T49	X2
A92	C86	C107	D32	E34	H25	J1	L21	M67	N13	O11	Q8	S42	T1	T51	73
B33	C87	C108	D37	E36	H29	J4	M17	M70	N15	P7	R7	S51	T11	V17	kilobaud
B35	C94	C109	D40	E37	I9	J6	M30	M73	N16	P9	R8	S52	T26	V19	
B43	C97	C111	D41	E38	I22	J12	M32	M74	N17	P21	R12	S55	T28	V20	

This month's ARTICLE WINNER (title page number): ____________

Name ______________________________

Address ______________________________

City ____________________ State ______ Zip ____________

KB 2/79

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 17 PETERBOROUGH NH 03458

POSTAGE WILL BE PAID BY ADDRESSEE

kilobaud MICROCOMPUTING

Peterborough NH 03458

ATTN: READER SERVICE

kilobaud MICROCOMPUTING

INSTANT INFORMATION

Please rush information on your products or services to the following address:

Your name

Number and street

City State Zip Code

Information requested from the February issue of Kilobaud.

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 17 PETERBOROUGH NH 03458

POSTAGE WILL BE PAID BY ADDRESSEE

kilobaud MICROCOMPUTING

Peterborough NH 03458

ATTN: READER SERVICE

kilobaud MICROCOMPUTING

books, etc.

BK1074 BASIC Computer Ga
Games—$7.50
BK1081 BASIC New 2nd Edi
—$4.95
BK1011 CMOS Cookbook—$
BK7322 Hobby Computers
Here!—$4.95
BK7325 How To Build A Mi
computer—And Really Un
stand It—$9.95
BK1131 Instant BASIC—$6.0

Instant Software From Kiloba
0001R Basic and Intermed
Lunar Lander—$7.95
0004R Beginner's Backgamn
and Keno—$7.95
0006P Mortgage with Pre
ment and Financier—$7.95
0005P Personal Weight Co
and Biorhythms—$7.95
0002R Space Trek II—$7.95
0014P Casino I/Blackjack-R
lette—$7.95
0015P Casino II/Craps—$7.9
0008R Electronics I—$7.95
0009R Golf/Cross-Out—$7.95
0007R Ham Package I—$7.95

BK1032 Introduction to Micro
cessors—$17.50 USA; $20 E
where
Kilobaud Coding Sheets
PD1001 Assembly/Machine L
guage—$2.39
PD1002 BASIC—$2.39
BK1037 Microprocessor Inter
ing Techniques—$9.95
BK1137 Microprocessor Lexi
—Acronyms and Definition
$2.95
BK1140 Programming In P
CAL—$9.95
Scientific Research Instrume
BASIC Software Library
LB1002 Vol. I—$24.95, LB1
Vol. II—$24.95, LB1004 Vo
—$39.95, LB1005 Vol. IV—$
LB1006 Vol. V—$9.95, LB1
Vol. VI—$49.95, LB1008 Vo
—$39.95
BK1012 The "Compulator" Bo
Building Super Calculator
Minicomputer Hardware W
Calculator Chips—$7.95
BK7340 The New Hobby Com
ers—$4.95
BK1063 TTL Cookbook—$8.95
BK1064 TVT Cookbook—$9.95
BK1071 What To Do After You
Return—$8.00
BK1172 Your Home Comput
$6.00